

Dr. Dobb's Journal

SOFTWARE TOOLS FOR ADVANCED PROGRAMMERS

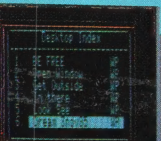
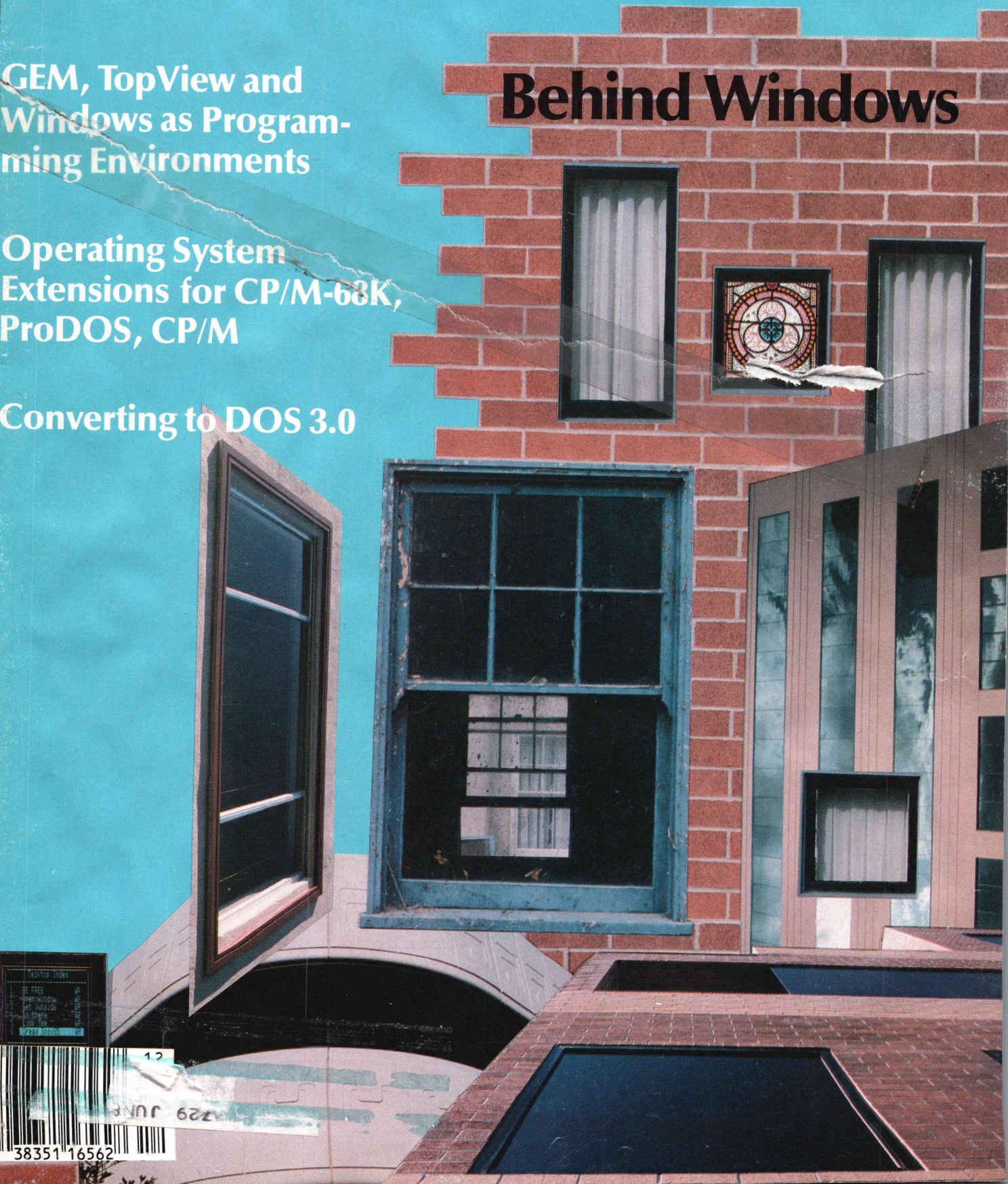
#110 DECEMBER 1985 \$2.95 (3.95 CANADA)

**GEM, TopView and
Windows as Program-
ming Environments**

**Operating System
Extensions for CP/M-68K,
ProDOS, CP/M**

Converting to DOS 3.0

Behind Windows



Breakthrough for C Programmers

H.E.L.P. Eliminates Every Bug known to Compilers ... As well as a few other species

H.E.L.P. is a completely interactive C programming environment with three innovative full-sized features that will revolutionize the way you write code.

A Clean Compile — Guaranteed!

Say Good-bye to all compiler-type errors. **H.E.L.P.**'s built-in program checker (which would embarrass LINT) not only hunts down bugs ... explains the proper syntax ... gives examples of usage ... but will even offer suggested corrections. If you want, **H.E.L.P.** will even make the corrections for you ... at the touch of a key.

H.E.L.P. also finds semantic errors as well as poor style and inefficiencies. You can even check the portability of your code!

Multi-Window Editing

Open as many windows as you want ... there's no limitation ... not even your own memory. Because **H.E.L.P.** uses a virtual memory system you can create programs larger than your machine capacity.

H.E.L.P.'s very powerful editor allows you the flexibility to work in several windows ... with several files at the same time.

Circle no. 165 on reader service card

Save Keystrokes

Hundreds of commands are bound to the keyboard to give you fast execution.

Always be in Control

Not only can you develop code in many windows at the same time, but you can show (and refer to) important definitions in one window while creating in another. Or open a window and keep notes about your program ... or type a memo ... or a letter.

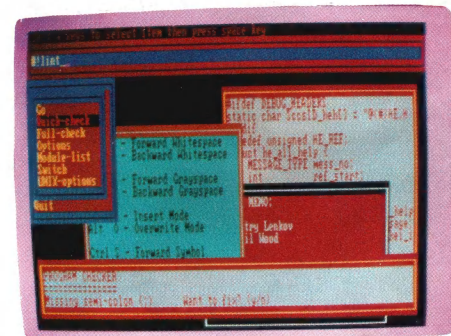
Increase your Productivity by 300% or More

If you are a novice programmer, you'll begin writing code like an advanced programmer much faster with **H.E.L.P.**

Just imagine what **H.E.L.P.** will do for the ADVANCED PROGRAMMER.

You'll have more time to become creative with your algorithm (since **H.E.L.P.** will make sure your code compiles the "first time at bat").

H.E.L.P. tracks every step you make. If you are not sure about a command, just press a key, and you'll get the kind of help you need.



Check These Features

- Multi-window environment
- Interactive program checking
- Check syntax, semantic, type usage, intermodule inconsistencies and portability
- Multi-file editing
- Intelligent help subsystem
- User-definable keyboard bindings
- Supports color and monochrome
- **H.E.L.P.** supports the full C Language

NOW IN MS-DOS

EVEREST
SOLUTIONS

Order now \$395

Everest Solutions, Inc.
3350 Scott Boulevard
Building 58
Santa Clara, CA 95051
(408) 986-8977



Optotech, Inc.

The 5¼ inch Optical Disk Drive Is Here!

Optical Disk Drive 5984

- 200 megabytes on a removable cartridge.
- Fast access read and write.
- For PCs and minis.
- Extensive interface software.
- Available Now.

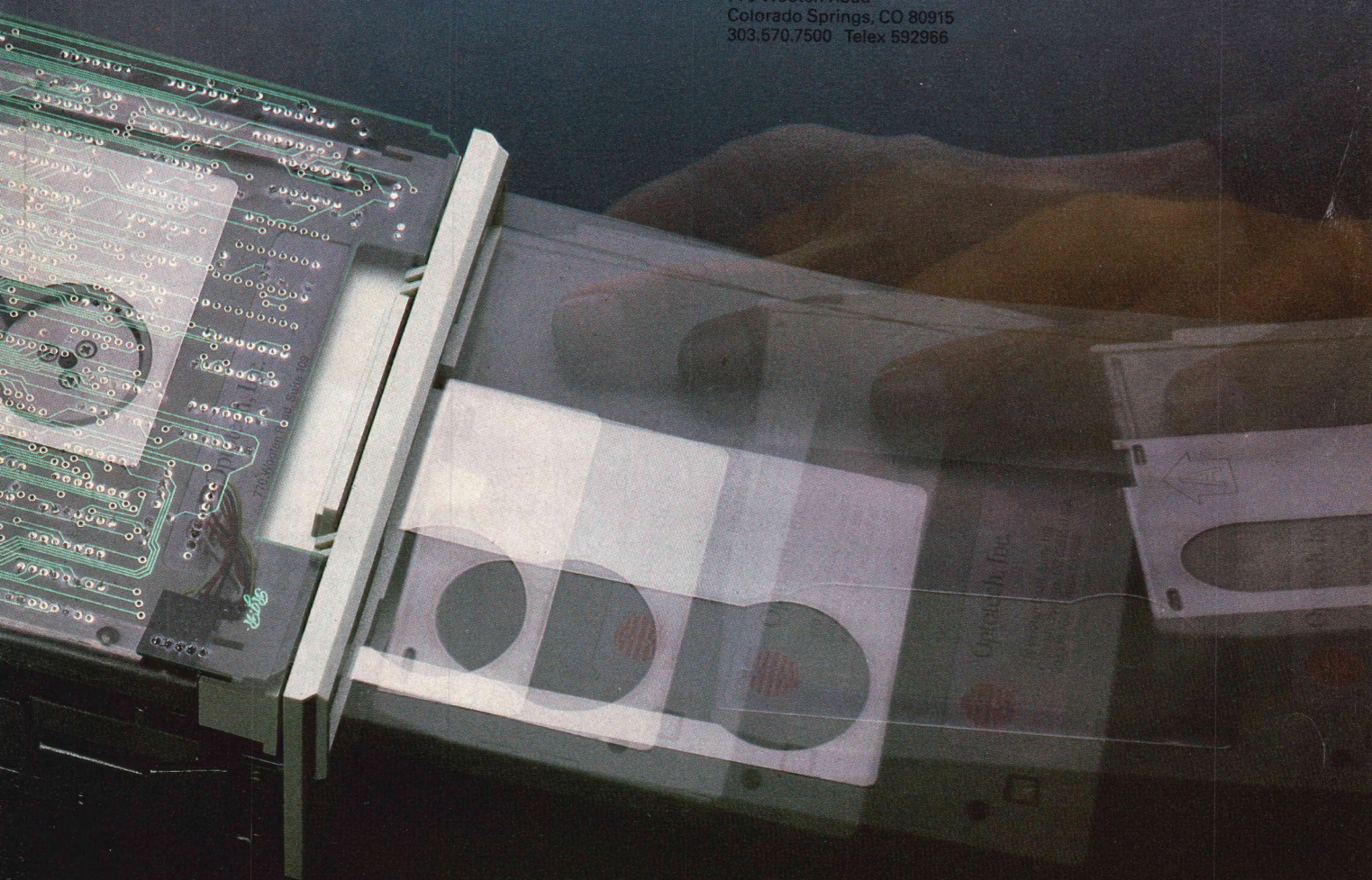
The Preferred Solution For:

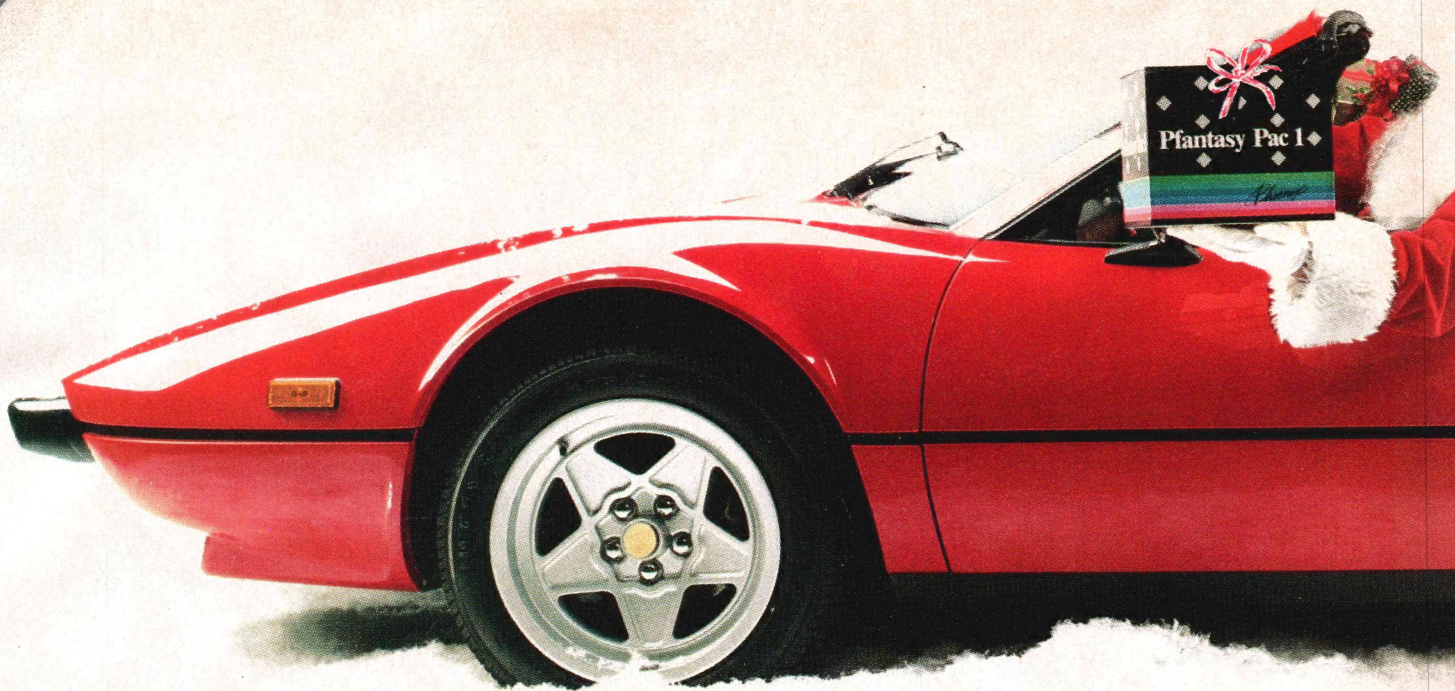
- Database—large, portable, indelible
and updatable
- Online Mass Storage—integral backup
- Imaging—capacity with removability



Optotech, Inc.

770 Wooten Road
Colorado Springs, CO 80915
303.570.7500 Telex 592966





Buy Yourself A Pfantasy Just ~~\$1295~~

Give your favorite programmer a holiday treat – Phoenix Pfantasy Pac1. Six of the best-engineered, highest performance programming packages on the market today. \$1800 sold separately. \$1295 as Pfantasy Pac1 with new software, and a unique on-line software update service. All at the one-time holiday price of \$995. The perfect gift for the knowledgeable programmer. You'll get:

...Plink™86 Plus, our new linkage editor, adds a new dimension to modular programming...

Cache overlays to reduce response time. Link programs much larger than possible with standard linkers. Merge object modules so you can package them together. Reload overlays automatically, so it's easier to design program structures. Or, create up to 4,095 overlays, nested up to 32 levels, in one or many files, on one or many disks.

...Plus, Pmate™, the text editor that works the way you do...

You can run Pmate in background while you perform another task, or call it up instantly while you're inside another program. With Pmate, you can reassign any command or key, so

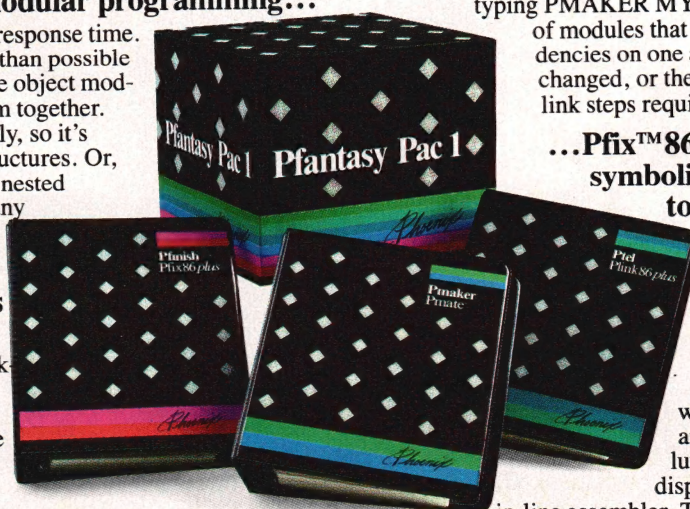
you can use keystrokes learned on other editors or make your keyboard Dvorak. And, you can assign complicated editing tasks to a keystroke, so you never have to re-key them.

...Pmaker™, the perfect program development manager...

Like UNIX™ Make and MMF, Pmaker updates and tracks modifications in your programs. With Pmaker, generating a final copy of your program is never more complicated than typing PMAKER MYFILE, regardless of the number of modules that make up MYFILE, their dependencies on one another, the number of modules changed, or the variety of compile/assemble/link steps required to generate the program.

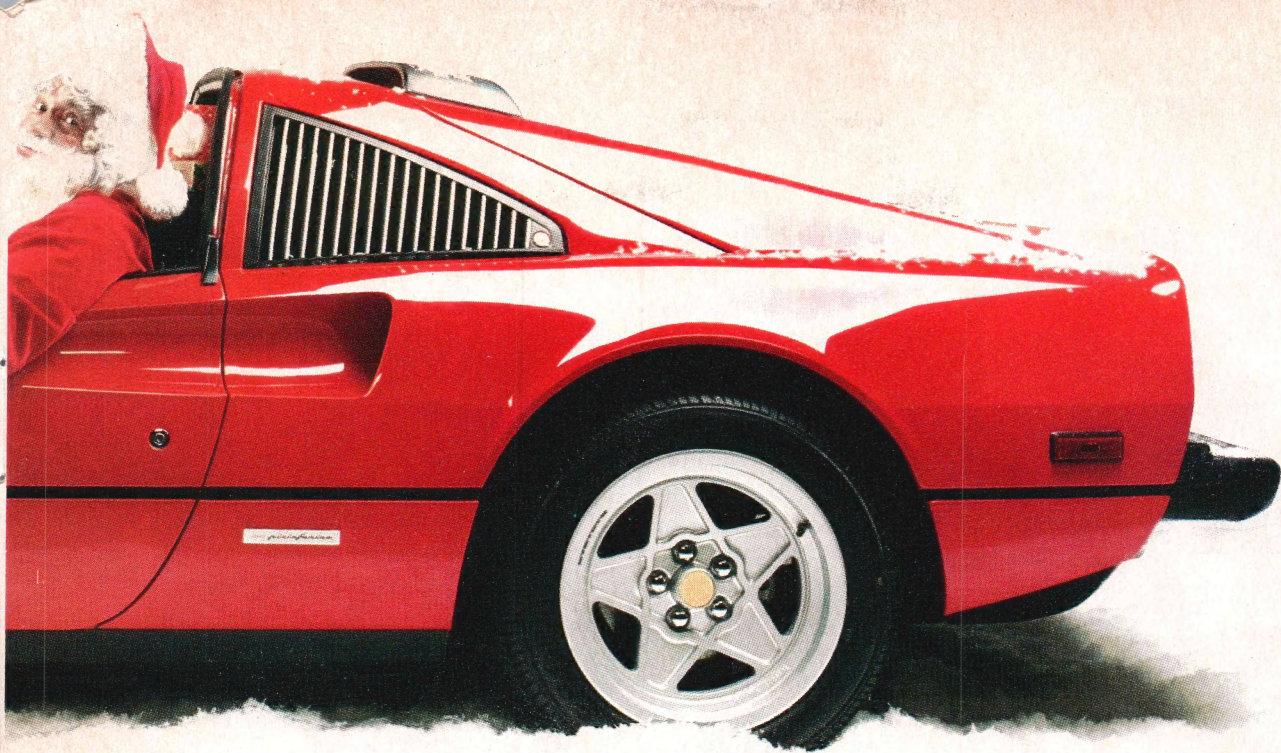
...Pfix™86 Plus, the most advanced symbolic debugger on the market today...

With its adjustable window display of source file, disassembled object, data area, stack, breakpoint settings, CPU and coprocessor registers, Pfix takes the guesswork from debugging. You can debug without a listing, since you can see and enter symbolic names or absolute addresses in breakpoints, data displays, expressions, or with the in-line assembler. The entire 80xx and 80xxx series of



*Price effective through December 31, 1985.

Circle no. 139 on reader service card.



Fantasy PacTM For The Holidays.

~~15.~~ \$995.*

16-bit processors is supported, along with the user-assignable string and numeric variables, up to 100-step trace-back, and debug log to disk or printer. Breakpoints can be set in the source file window, whose display can be synchronized with the disassembled object during tracing; symbols can be added incrementally during the debug session; and the disassembly written to disk.

...PfinishTM, the program that maximizes your program's efficiency...

Pfinish helps you turn your beta-test product into a software work of art. It analyzes your program or the entire operating environment during execution, and produces reports which tell you how much time was spent in each routine or interrupt, who called it, how many times, and much more. Wasteful, inefficient, or non-optimal areas of code become immediately apparent, whether they're in your code, the compiler library, or in the operating system routines themselves.

...PtelTM gets the lead out of binary file transfer...

You get error-free file transfer and access to mainframes, minis, and micros. ASCII. XModem. Modem7. Telink and Kermit. MS-DOS[®] 2.x or 3.x. With Ptel. You can transfer 8-bit binary files over a 7-bit data path with Kermit. Masked file-name transfers with XModem. Or, transfer whole sub-directories with a single command using Telink. Ptel keeps track of the original file size and creation date. Ptel also offers a script language, backward scrolling, and the ability to handle DOS commands from inside the program.

...And, Phoenix's new on-line software update service. Free to all registered Pfantasy Pac1 owners.

Pfantasy Pac1 software works in MS-DOS/PC DOS environments, is completely language-independent, and is available for the IBM[®] PC, XT,TM AT,TM and compatibles, the Wang[®] PC, the TI Business-Pro,TM and the Tandy[®] 2000.

So, spoil yourself a little. Get Pfantasy Pac1, and make your favorite programmer's dream come true.

For more information contact: Phoenix Computer Products Corporation, 320 Norwood Park South, Norwood, MA 02062. Or call: (800) 344-7200. In Massachusetts (617) 762-5030.

Programmers' PfantasiesTM
by

Phoenix

Programmers' Pfantasies, Pfantasy Pac, Pmaker, and Pfinish are trademarks of Phoenix Computer Products Corporation. Pmate, Pfix86 Plus, Plink86 Plus, and Ptel are trademarks of Phoenix Software Associates Ltd. MS-DOS is a registered trademark of Microsoft Corporation. Wang is a registered trademark of Wang Laboratories, Inc. Tandy is a registered trademark of Tandy Corporation. UNIX is a trademark of AT&T Bell Laboratories. IBM is a registered trademark. XT and AT are trademarks of International Business Machines Inc. TI Business-Pro is a trademark of Texas Instruments Incorporated.

Dr. Dobb's Journal

Editorial

Editor-in-Chief Michael Swaine
Managing Editor Frank DeRose
Editorial Assistant Sara Noah
Contributing Editors Robert Blum,
Dave Cortesi,
Ray Duncan,
Allen Holub

Copy Editor Vince Leone
Typesetter Jean Aring

Production

Production Manager Bob Wynne
Art Director Shelley Rae Doeden
Production Assistant Alida Hinton
Cover Artist Tom Upton

Circulation

Sub. Fulfillment Mgr. Stephanie Barber
Subscription Mgr. Maureen Kaminski
Book Marketing Mgr. Jane Sharninghouse
Single Copy Sales Mgr. Stephanie Barber

Administration

Finance Manager Sandra Dunie
Business Manager Betty Trickett
Accounts Payable Supv. Mayda Lopez-Quintana
Accounts Payable Asst. Denise Giannini
Billing Coordinator Laura Di Lazzaro
Accountant Marilyn Brown
Adm. Coordinator Kobi Morgan

Advertising

Advertising Director

Shawn Horst (415) 424-0600

Advertising Sales

Walter Andrzejewski (617) 567-8361

Lisa Boudreau (415) 424-0600

Beth Dudas (714) 643-9439

Michele Beaty (317) 875-8093

Systems Manager Ron Copeland

Administrative Manager Anna Kittleson

M&T Publishing, Inc.

Chairman of the Board Otmar Weber
Director C.F. von Quadt
President Laird Foshay

Dr. Dobb's Journal (USPS 307690) is published monthly by M&T Publishing, Inc., 2464 Embarcadero Way, Palo Alto, CA 94303, (415) 424-0600. Second class postage paid at Palo Alto and at additional entry points.

Address correction requested. Postmaster: Send Form 3579 to *Dr. Dobb's Journal*, P.O. Box 27809, San Diego, CA 92128. **ISSN 0278-6508**

Customer Service: For subscription problems call: outside CA 800-321-3333; within CA 619-485-9623 or 566-6947. For book, back issue, or disk order problems call 415-424-1474.

Subscription Rates: \$25 per year within the United States. Foreign subscription rates: \$52.00 for airmail, \$42.00 for surface mail. Foreign subscriptions must be pre-paid in U.S. Dollars, drawn on a U.S. Bank.

Foreign Distributor: Worldwide Media Service, Inc., 386 Park Ave. South, New York, NY 10016, (212) 686-1520 TELEX: 620430 (WUI)

Entire contents copyright © 1985 by M&T Publishing, Inc. unless otherwise noted on specific articles. All rights reserved.



People's Computer Company

Dr. Dobb's Journal is published by M&T Publishing, Inc. under license from People's Computer Company, 2682 Bishop Dr., Suite 107, San Ramon, CA 94583, a non-profit, educational corporation.

December 1985
Volume 10, Issue 12

CONTENTS

In this Issue

This is our second annual operating systems issue. We've tried to present a variety of tools for extending the performance of several operating systems. We looked for techniques and programs that would make software development easier: a debugging tool, techniques for expanding the available memory for programs, a conditional instruction tool for CP/M-68K.

We also took a look at three commercial products that claim to extend the performance of the MS DOS operating system: DRI's GEM, IBM's Top-View, and Microsoft's Windows. We talked to developers of each product, asking what programmers need to know to develop applications compatible with these operating environments and why a programmer might want to do such a thing.

Look closely at the issue next month; it won't look quite the same. Our art director has introduced some changes inside and outside the magazine that we think you'll like.

We don't think you'll like hearing that Dr. Dobb's Clinic is closing with this issue. We don't like telling you. The resident intern, Dave Cortesi, has gone off to pursue interests having nothing to do with software. We'll miss Dave's gentle wit. We'd like to reopen the Clinic, but we haven't found anyone yet to fill Dave's shoes, and until we do, we won't.

Next month: 68000 programming.

STATEMENT OF OWNERSHIP, MANAGEMENT, AND CIRCULATION

(Act of August 12, 1970, Section 3685, Title 39, United States Code)

1. Title of Publication: Dr. Dobb's Journal, Publication No. 307690.
2. Date of Filing: October 12, 1985.
3. Frequency of Issue: Monthly (12 issues, \$25).
4. Location of Known Office of Publication: 2464 Embarcadero Way, Palo Alto, CA 94303.
5. Location of Headquarters of General Business Offices of the Publishers: 2464 Embarcadero Way, Palo Alto, CA 94303.
6. Names and Addresses of Publisher, Editor, and Managing Editor: Publisher, Laird Foshay, 2464 Embarcadero Way, Palo Alto, CA 94303. Editor, Michael Swaine, 2464 Embarcadero Way, Palo Alto, CA 94303. Managing Editor, Frank DeRose, 2464 Embarcadero Way, Palo Alto, CA 94303.
7. Owner: Otmar Weber, 2464 Embarcadero Way, Palo Alto, CA 94303; and C. F. Von Quadt, 2464 Embarcadero Way, Palo Alto, CA 94303.
8. Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities: Markt & Technik, Hans-Pinsel-Strasse 2 8013 Haar bei Munich W. Germany.
9. Extent and Nature of Circulation:
 - A. Total number of copies printed: Average number of copies each issue during the preceding 12 months: 41,693. Actual number of copies of single issue published nearest to filing date: 45,037.
 - B. Paid Circulation. 1. Sales through dealers and carriers, street vendors, and counter sales. Average number of copies each issue during preceding 12 months: 8,707. Actual number of copies of single issue published nearest to filing date: 7,815. 2. Mail subscrip-

tions. Average number of copies each issue during preceding 12 months: 26,568. Actual number of copies of single issue published nearest to filing date: 30,417.

C. Total Paid Circulation. Average number of copies each issue during preceding 12 months: 35,275. Actual number of copies of single issue published nearest to filing date: 38,232.

D. Free distribution by mail, carrier, or other means, samples, complimentary, and other free copies. Average number of copies each issue during preceding 12 months: 2,353. Actual number of copies of single issue published nearest to filing date: 3,305.

E. Total distribution. Average number of copies each issue during preceding 12 months: 37,628. Actual number of copies of single issue published nearest to filing date: 41,537.

F. Copies not distributed. 1. Office use, left over, unaccounted, spoiled after printing. Average number of copies each issue during preceding 12 months: 150. Actual number of copies of single issue published nearest to filing date: 150. 2. Returns from news agents. Average number of copies each issue during preceding 12 months: 3,915. Actual number of copies of single issue published nearest to filing date: 3,350.

G. Total. Average number of copies each issue during preceding 12 months: 41,693. Actual number of copies of single issue published nearest to filing date: 45,037.

I certify that the statements made by me above are correct and complete.

Laird Foshay, Publisher

Dr. Dobb's Journal

ARTICLES

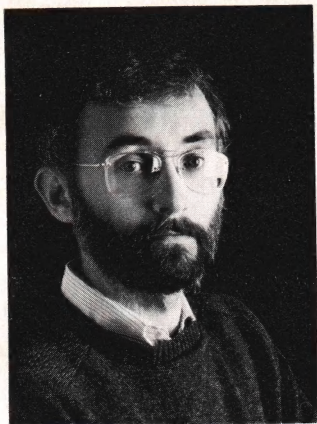
- Windowing Operating Environments** 28 We talked to the developers behind GEM, TopView, and Windows to find out what these operating environments could mean to software developers. **Reader Ballot No. 192.**
by Michael Swaine
- BANKSWAP: A Banked Memory Debugging Tool for CP/M Plus** 34 CP/M Plus can address a megabyte, but debugging beyond the first 64K bank is a problem. **Reader Ballot No. 193.**
by Albert S. Woodhull
- Adding a Copy Command to ProDOS** 54 COPY solves the problem of copying files between volumes without losing a BASIC program in memory. **Reader Ballot No. 194.**
by Shawn Day
- CP/M-68K Conditional Batch Processing** 70 Efficient programming requires tools for automatically compiling, assembling, and linking with termination on error. IF.68K extends CP/M-68K in that direction. **Reader Ballot No. 195.**
by Roger E. Donais
- ProDOS and the RAM Disk** 84 Here's how one programmer developed a RAM-disk driver for ProDOS. **Reader Ballot No. 196.**
by Alfred Steele

COLUMNS

- Dr. Dobb's Clinic** 14 The Clinic closes its doors. **Reader Ballot No. 190.**
by D.E. Cortesi
- C Chest** 16 Support routines for a new shell for MS DOS; An fgets that edits. **Reader Ballot No. 191.**
by Allen Holub
- 16-Bit Software Toolbox** 102 Readers flame on Microsoft Assembler; notes on converting to MS DOS 3.0. **Reader Ballot No. 197.**
by Ray Duncan
- Mac Toolbox** 110 Concluding the MacSCSI interface series. **Reader Ballot No. 198.**
by John Bass
- CP/M Exchange** 114 In honor of our focus on operating system extensions, our guest columnist tells of a product that makes CP/M a big operating system. **Reader Ballot No. 199.**
by W. E. Wilson

DEPARTMENTS

- Editorial** 6
- Letters** 8
- Advertiser Index** 128



It's been a litigious autumn in the software industry. At press time, Apple had forced DRI into changing the GEM Desktop; Apple lawyers were eyeing Atari and Microsoft; and Atari, anxious to get its software into ROM, was reportedly ready to challenge Apple. Apple's contention that its copyright protected not only the expression of algorithms in code but also the visual expression of the desktop (and implicitly that it had invented that desktop) had yet to be tested in court.

Adapso had been running ads attempting to educate software users about the criminal nature of unauthorized copying of software. Several software publishers had brought suit against customers to aid in the pedagogical process—all in reaction to a perceived loss of (potential) revenues and all very reminiscent of a campaign by the author of MITS BASIC a decade earlier. It wasn't a good tactic then to call your (potential) customers thieves and probably isn't now, but how are authors' and publishers' investments in software to be protected without alienating users?

Vault, a manufacturer of a copy-protection product, concluded that a Canadian company must have illegally disassembled Vault's code in order to produce its unlocking product; so Vault found a state in which it thought it could get a decision in its favor and sued. Vault lost, but one decision in a Louisiana court hardly dissipates the moral ambiguity that surrounds the entire copy-protection issue. Do we really want this escalating arms race between protectors and crackers?

The Vault case dealt with trade secrets. Copyright applies to works that are in some manner made public. No author of a book would object to someone analyzing his work in order to understand it—disassembling it, if you will. Trade secrets, on the other hand, are necessarily secret. Whom do trade secrets protect, and whom does copyright protect? Many software publishers treat the two protection methods as compatible, but does it really make sense that you can protect as a trade secret something protected under copyright law? As a book reviewer I could, under the fair-use interpretation of copyright law, excerpt and analyze a work of literature. Could I, as a software reviewer, disassemble and publish portions of commercial programs to show the workings of key algorithms?

These questions suggest, at the very least, that not everyone is of one mind regarding the protection of intellectual property. They also suggest something more alarming: that precedents are being set and opinions are being molded by bluster and threat, by cases that never come to court, and by venue-choosing ploys that take advantage of differing levels of understanding of the issues in different states. It shouldn't be that way.

You could do something about it. You probably know more about the issues than do most journalists, judges, or jurors. And I suspect you may have more empathy for the user than does the average software publisher, who often has stockholders to answer to. I urge you to make your expertise available to those making decisions with long-term implications. Educate lawyers and editors.

Become an expert witness for rational software law.

Michael Swaine

Michael Swaine



The C for Microcomputers

PC-DOS, MS-DOS, CP/M-86, Macintosh, Amiga, Apple II, CP/M-80, Radio Shack, Commodore, XENIX, ROM, and Cross Development systems

MS-DOS, PC-DOS, CP/M-86, XENIX, 8086/80x86 ROM

Manx Aztec C86

"A compiler that has many strengths ... quite valuable for serious work"

Computer Language review, February 1985

Great Code: Manx Aztec C86 generates fast executing compact code. The benchmark results below are from a study conducted by Manx. The Dhrystone benchmark (CACM 10/84 27:10 p1018) measures performance for a systems software instruction mix. The results are without register variables. With register variables, Manx, Microsoft, and Mark Williams run proportionately faster, Lattice and Computer Innovations show no improvement.

	Execution Time	Code Size	Compile/Link Time
Dhrystone Benchmark			
Manx Aztec C86 3.3	34 secs	5,760	93 secs
Microsoft C 3.0	34 secs	7,146	119 secs
Optimized C86 2.20J	53 secs	11,009	172 secs
Mark Williams 2.0	56 secs	12,980	113 secs
Lattice 2.14	89 secs	20,404	117 secs

Great Features: Manx Aztec C86 is bundled with a powerful array of well documented productivity tools, library routines and features.

Optimized C compiler	Symbolic Debugger
AS86 Macro Assembler	LN86 Overlay Linker
80186/80286 Support	Librarian
8087/80287 Sensing Lib	Profiler
Extensive UNIX Library	DOS, Screen, & Graphics Lib
Large Memory Model	Intel Object Option
Z (vi) Source Editor -c	CP/M-86 Library -c
ROM Support Package -c	INTEL HEX Utility -c
Library Source Code -c	Mixed memory models -c
MAKE, DIFF, and GREP -c	Source Debugger -c
One year of updates -c	CP/M-86 Library -c

Manx offers two commercial development systems, Aztec C86-c and Aztec C86-d. Items marked -c are special features of the Aztec C86-c system.

Aztec C86-c Commercial System	\$499
Aztec C86-d Developer's System	\$299
Aztec C86-p Personal System	\$199
Aztec C86-a Apprentice System	\$49

All systems are upgradable by paying the difference in price plus \$10.

Third Party Software: There are a number of high quality support packages for Manx Aztec C86 for screen management, graphics, database management, and software development.

C-tree \$395	Greenleaf \$185
PHACT \$250	PC-lint \$98
HALO \$250	Amber Windows \$59
PRE-C \$395	Windows for C \$195
WindScreen \$149	FirstTime \$295
SunScreen \$99	C Util Lib \$185
PANEL \$295	Plink-86 \$395

MACINTOSH, AMIGA, XENIX, CP/M-68K, 68k ROM

Manx Aztec C68k

"Library handling is very flexible ... documentation is excellent ... the shell a pleasure to work in ... blows away the competition for pure compile speed ... an excellent effort."

Computer Language review, April 1985

Aztec C68k is the most widely used commercial C compiler for the Macintosh. Its quality, performance, and completeness place Manx Aztec C68k in a position beyond comparison. It is available in several upgradable versions.

Optimized C	Creates Clickable Applications
Macro Assembler	Mouse Enhanced SHELL
Overlay Linker	Easy Access to Mac Toolbox
Resource Compiler	UNIX Library Functions
Debuggers	Terminal Emulator (Source)
Librarian	Clear Detailed Documentation
Source Editor	C-Stuff Library
MacRam Disk -c	UniTools (vi, make, diff, grep) -c
Library Source -c	One Year of Updates -c

Items marked -c are available only in the Manx Aztec C86-c system. Other features are in both the Aztec C86-d and Aztec C86-c systems.

Aztec C68k-c Commercial System	\$499
Aztec C68d-d Developer's System	\$299
Aztec C68k-p Personal System	\$199
C-tree database (source)	\$399
AMIGA, CP/M-68k, 68k UNIX	call

Apple II, Commodore, 65xx, 65C02 ROM

Manx Aztec C65

"The AZTEC C system is one of the finest software packages I have seen"

NIBBLE review, July 1984

A vast amount of business, consumer, and educational software is implemented in Manx Aztec C65. The quality and comprehensiveness of this system is competitive with 16 bit C systems. The system includes a full optimized C compiler, 6502 assembler, linkage editor, UNIX library, screen and graphics libraries, shell, and much more. The Apple II version runs under DOS 3.3, and ProDOS, Cross versions are available.

The Aztec C65-c/128 Commodore system runs under the C128 CP/M environment and generates programs for the C64, C128, and CP/M environments. Call for prices and availability of Apprentice, Personal and Developer versions for the Commodore 64 and 128 machines.

Aztec C65-c ProDOS & DOS 3.3	\$399
Aztec C65-d Apple DOS 3.3	\$199
Aztec C65-p Apple Personal system	\$99
Aztec C65-a for learning C	\$49
Aztec C65-c/128 C64, C128, CP/M	\$399

Distribution of Manx Aztec C

In the USA, Manx Software Systems is the sole exclusive distributor of Aztec C. Any telephone or mail order sales other than through Manx are unauthorized.

Manx Cross Development Systems

Cross developed programs are edited, compiled, assembled, and linked on one machine (the HOST) and transferred to another machine (the TARGET) for execution. This method is useful where the target machine is slower or more limited than the HOST, Manx cross compilers are used heavily to develop software for business, consumer, scientific, industrial, research, and educational applications.

HOSTS: VAX UNIX (\$3000), PDP-11 UNIX (\$2000), MS-DOS (\$750), CP/M (\$750), MACINTOSH (\$750), CP/M-68k (\$750), XENIX (\$750).

TARGETS: MS-DOS, CP/M-86, Macintosh, CP/M-68k, CP/M-80, TRS-80 3 & 4, Apple II, Commodore C64, 8086/80x86 ROM, 68xxx ROM, 8080/8085/Z80 ROM, 65xx ROM.

The first TARGET is included in the price of the HOST system. Additional TARGETS are \$300 to \$500 (non VAX) or \$1000 (VAX).

Call Manx for information on cross development to the 68000, 65816, Amiga, C128, CP/M-68K, VRTX, and others.

CP/M, Radio Shack, 8080/8085/Z80 ROM

Manx Aztec CII

"I've had a lot of experience with different C compilers, but the Aztec C80 Compiler and Professional Development System is the best I've seen."

80-Micro, December, 1984, John B. Harrell III

Aztec C II-c (CP/M & ROM)	\$349
Aztec C II-d (CP/M)	\$199
C-tree database (source)	\$399
Aztec C80-c (TRS-80 3 & 4)	\$299
Aztec C80-d (TRS-80 3 & 4)	\$199

How To Become an Aztec C User

To become an Aztec C user call 1-800-221-0440 or call 1-800-832-9273 (800TEC WARE). In NJ or outside the USA call 201-530-7997. Orders can also be telexed to 4995812.

Payment can be by check, COD, American Express, VISA, Master Card, or Net 30 to qualified customers.

Orders can also be mailed to Manx Software Systems, Box 55, Shrewsbury, NJ 07701.

How To Get More Information

To get more information on Manx Aztec C and related products, call 1-800-221-0440, or 201-530-7997, or write to Manx Software Systems.

30 Day Guarantee

Any Manx Aztec C development system can be returned within 30 days for a refund if it fails to meet your needs. The only restrictions are that the original purchase must be directly from Manx, shipped within the USA, and the package must be in resalable condition. Returned items must be received by Manx within 30 days. A small restocking fee may be required.

Discounts

There are special discounts available to professors, students, and consultants. A discount is also available on a "trade in" basis for users of competing systems. Call for information.

MANX

To order or for information call:

800-221-0440

UNIX is a registered TM of Bell Laboratories. Lattice TM Lattice Inc. C-draw TM Faircom Inc. PHACT TM PHACT ASSOC. CI Optimizing C86 TM Computer Innovations Inc. MACINTOSH APPLE TM APPLE INC. PINK PLINK 86 TM PHOENIX HALO TM Media Cybernetics Inc. C-tree TM C-tree Software. WindScreen SunScreen TM SunSoft. PANEL TM Roundhill Computer Systems Ltd. WINDOWS FOR C TM Creative Solutions. XENIX MS TM MICROSOFT INC. CP/M TM DRI AMIGA C64 C128 TM COMMODORE INC.

Circle no. 108 on reader service card.



Small-C

Dear DDJ,

The Small-C problem reported by Gregor Owen (Letters, July 1985) has already been identified, fixed, and reported to Small-C users on my mailing list. It also appears in the "Small-C Update" article (DDJ, August 1985).

As it turns out, Owen's change does not address the real problem and introduces serious problems of its own. It is generally not a good idea simply to remove functions from a program even if they are referenced only once. Most programmers have reasons, other than the introduction of bugs, for placing even poorly documented functions in programs.

The function `result()` serves the vital role of causing the expression analyzer to note for binary operators whether or not the result is an address, and if so whether it is a character or integer address. Removing it will cause the analyzer always to assume the attributes of the left operand for the operator value. For example, "`addr1 - addr2`" should yield an integer result indicating the number of objects (bytes or words) between the two addresses. Without `result()`, Small-C will think it has another address, and that has ramifications for the further process of expression evaluation.

The real problem was that the analyzer was losing track of the attributes of the `lvalue` for `?` operators when the right side evaluated to an address. The correction was to make `hier1()` (in CC31.C) remember two of the `lvalue` attributes for use at store time. This is done as follows:

1. define the local integer array `lval3[2]` after `lval2[8]`.
2. between "`if(k==0)...`" and "`if(lval[1])...`" insert

```
lval3[0] = lval[0];
lval3[1] = lval[1];
```

3. change the call to `store()` to read
`store(lval3);`

Owen's change appeared to work because in his test case the error he introduced caused the compiler to sidestep the original problem. The troublesome statement was:

```
lptr += r - bytes
```

where "r" was an integer and "bytes" was an array name (yields the array address). As mentioned above, the removal of `result()` would cause "r - bytes" to yield an integer result, thereby avoiding the original problem.

By the way, one might question the meaning of an integer minus an address. K&R point out on page 99 that such expressions are illegal.

Jim Hendrix
Rt. 1, Box 585
Oxford, MS 38655

Information Age Issues

Dear DDJ,

Please continue publishing articles like Dean Gengle's "Information Age Issues" (June 1985). Such ethical issues are too important to leave in the hands of a systems designer.

I must, however, take issue with Gengle's comment that "no one is talking about our rights to access information that is *not* private and that was collected legally at taxpayers' expense. Census data and Library of Congress information come to mind."

Your readers should know that both the American Library Association Government Documents Roundtable and the Special Libraries Association Government Relations Committee have indeed been talking

about it. In addition, the Joint Committee on Printing of the Congress, which oversees the Government Printing Office, has been very concerned that material produced in machine-readable form by Federal agencies has not been made available to the GPO for distribution to the public through the depository libraries because it is not "printed."

Librarians protested strongly the distribution of census data in microfiche-only format because such data is not readily accessible to the poor or illiterate. Distribution on magnetic tape is a great service to those who can afford to have the tapes processed but may harm those who depend on such data for distribution of entitlement program benefits.

Write to your representatives and urge them to support the efforts of the Joint Committee on Printing, the ALA, and SLA.

I'm not sure what Gengle is referring to by "Library of Congress information." He may be suffering from the misinformation provided by an episode of the television program "Whiz Kids" in which the elfin heroes broke into the Library of Congress computers and did a homework assignment. I doubt if there are any files at the LC, or anywhere else for that matter, that could produce the results obtained. In fact, the files at the LC are bibliographic, containing references to literature, but not full texts—at least not yet. Not only are these bibliographic files available to the public at the LC but they are available on magnetic tape to libraries and other institutions throughout the world. Many libraries located in this country access those records through OCLC, a bibliographic utility with over 11 million records. Some agencies do apparently try to hide

Wendin® puts Power and Punch in your PC

ORDER NOW!
Prices on PCUNIX™ and
PCVMS™ go up to \$99
effective January 1, 1986

Operating System Toolbox™ only \$9900

Operating System Toolbox™ is a software construction set that enables you to build your own multitasking, multiuser operating system. All you need is your creativity and imagination to write a shell and link it with the Toolbox. What you receive for your effort is an operating system that interacts with the way **YOU** work, performs the jobs that **YOU** need in the way **YOU** want them done. Toolbox runs MS-DOS and PC-DOS programs with full source code written in Microsoft "C", organized into seven basic modules, all on disk. The package includes both source and object code for each module (just in case you don't have Microsoft "C"), plus a fantastic, in-depth, step-by-step instruction manual on how to build your personal operating system.

The best part of the news is the price — so low, you'll want to order today, just to keep up with the state of the art in operating systems.

PCUNIX™ only \$4900

PCUNIX™ is powerful! It is a true multitasking, multiuser operating system similar to the popular UNIX® operating system from AT&T. This first release includes 30 utilities, designed to make your program development a snap.

PCUNIX™ is versatile! It runs MS-DOS and PC-DOS programs, so that you'll never have to buy another set of development tools! Simply use your existing compilers and linkers. You need nothing more! As with the development of all of our Operating Systems, Wendin has used the Wendin Operating System Toolbox™ to construct PCUNIX™. Because of this, your programs will have access to the over 70 enhanced system services found in the Wendin Operating System Toolbox!™

PCUNIX™ is complete! Full source code to the shell and all utilities, written in "C", is included with this incredible package. And if you would like the source to Wendin's Personal Operating System Kernel, you may also want to purchase the Wendin Operating System Toolbox.™

PCVMS™ only \$4900

PCVMS™ is Wendin's version of the popular VAX/VMS operating system, which was developed by Digital Equipment Corporation for their line of VAX computers. PCVMS™ turns your IBM-PC, XT, AT, or true compatible into a supercharged, multitasking, multiuser workstation that runs MS-DOS and PC-DOS programs.

As with the development of all of our Operating Systems, Wendin has used the Operating System Toolbox™ to construct PCVMS, allowing your programs access to the over 70 enhanced system services available in the Toolbox.

PCVMS™ comes with full source code to the PCVMS™ shell and its utilities, all written in "C". If you also want the source to Wendin's Personal Operating System Kernel, you may want to purchase the Wendin Operating System Toolbox.™

ORDER HOTLINE
509/235-8088
Master Card & Visa

Foreign orders inquire about shipping.
Domestic orders add \$3.50/1st item, \$1.00 each additional item for shipping, handling and insurance.

Washington residents add 7.8% sales tax.



WENDIN®

BOX 266
CHENEY, WA 99004

The people who make quality
software tools affordable.

DEALER INQUIRIES WELCOME!

PC-DOS is a trademark of IBM
MS is a trademark of Microsoft
Unix is a trademark of AT&T
VAX/VMS is a trademark of Digital Equipment Corporation

Wendin and XTC are Registered Trademarks of Wendin, Inc.

PCVMS, PCUNIX, and Operating System Toolbox are Trademarks of Wendin, Inc.

XTC® only \$9900

XTC® is the world's first multitasking editor for the IBM-PC. It also runs on IBM-XT and IBM-AT computers, as well as true compatibles. Designed BY programmers FOR programmers, XTC is the ultimate editing tool for software developers using C, PASCAL, ASSEMBLY, BASIC, FORTRAN, and other languages.

Why is XTC® the ultimate tool for editing in your development environment? Because it has powerful features like MULTITASKING MACROS, WINDOWS, TEXT BUFFERS, UNDO N TIMES, MACRO PROGRAMMING CONTROL STRUCTURES and VARIABLES, as well as blinding speed that leaves other editors in the stone age.

XTC® comes with full source code written in Microsoft Pascal, on 2 DSDD disks.

their information from the public, but the Library of Congress certainly does not.

Bruce B. Cox
Documents Librarian and
Automation Committee
Linda Hall Library
5109 Cherry Street
Kansas City, MO 64110

C Compiler Review

Dear DDJ,

I read with great interest your August 1985 review of MS DOS C compilers. I have two comments I would like to interject into the debate.

First, regarding the review of the documentation—this is always a touchy area, being rather subjective

in impact—I am not convinced that a C compiler manual should include what is essentially a rehash of the standard K&R syntax chapters nor a rehash of the standard Unix manual's chapters 2 and 3. That is, unless a compiler and run-time library deviate from these accepted standards, I see no reason to repeat them. It only serves to confuse. As an example, I find the Lattice C documentation difficult to use because I generally expect it to be similar to the Unix manual. It isn't, and I always get lost trying to find something.

It is much better simply to include a "pointer" to the appropriate documentation with the caveat that any functions and/or syntactic notes in-

cluded in the product's manual imply a deviation from the norm. Sort of like a "search path" if you will. One uses the Unix and K&R manuals unless the product's manual overrides them.

Second, the authors did not include as a criterion of judgement the number of other products that support a particular compiler. For instance, although Lattice C does not necessarily produce the tightest code nor run the quickest, it is important that one can obtain interfaces to major products. The Oracle DBMS includes a host-language interface (HLI) to Lattice C, and a HLI to dBASE II and III can be obtained from Lattice. Such interfaces are of great significance, and in these systems the choice of compiler will hinge entirely upon that single criterion. It is better, in other words, to put up with slightly slower execution than to have to generate in house an entire nontrivial HLI.

Otherwise, I was pleased with the review—it was obviously a strong effort. It is gratifying to this C and Unix nut to see the wide interest in these tools for programmers.

Brian Jay Wu
P.O. Box 203
Newbury Park, CA 91320

Dear DDJ,

When I heard that your magazine was going to do a review of C compilers and interpreters, I couldn't wait to read it.

The cover for the August 1985 issue claims that the review of C compilers is definitive. I think the review is marginally useful and generally misleading. Information important to the potential user was omitted. A list of considerations relating to the usefulness of the various C compilers follows:

1. Good luck to anyone who purchases the Microsoft compiler Version 3.0 without a hard disk! The four passes of the compiler total 290K, the CL.EXE driver takes 27K, the linker takes 41K, and the libraries needed total at least 120K. Add it up and you get a whopping 478K.

It is becoming common to run a compiler from inside your editor.



110 MB disk/tape subsystem for XENIX

Finally, a complete XENIX subsystem for the AT.

Disk Features

- 30, 40, 55, 72, 118 Megabytes (formatted)
- Combine drives with each other or existing drive
- 25 milliseconds average access time
- Simplified installation
- Necessary file modifications done automatically

Tape Features

- 60 Megabyte 1/4 inch cartridge
- Standard XENIX commands (cpio, tar, dd, etc.)
- Fully integrated driver software

Subsystem Features

- Entire subsystem fits inside the AT
- External version with 6 expansion slots available (pictured)
- One year factory warranty



Emerald
Systems Corporation

Mainframe Storage for Micros

4757 Morena Boulevard
San Diego, CA 92117
(619) 270-1994
Telex 323458 EMERSYS
EasyLink 62853804

Emerald & Mainframe Storage for Micros™ Emerald Systems Corp.

If you're tired of waiting, you're using the wrong file manager.



Be fast. Btrieve.™

If batch jobs and reports turn waiting time to nighttime, then wake up! You're using the wrong file manager.

Btrieve™ file management hates waiting as much as you do. It's written in assembly language especially for the IBM PC. And based on b-tree file indexing, with automatic balancing for access speed that won't degrade as your database grows. With Btrieve, your applications always run fast. So you'll be out the door faster.

The standard for networking. When your application requires multi-user file sharing, Btrieve/N (network version) sets the standard for the industry's most popular LANs: IBM's PC Network, Netware, Omninet, PC Net, Ether Series, Nestar, and NetOne. Btrieve/N offers *safe* network file management that coordinates simultaneous updates and prevents lost data.

Automatic file recovery. Btrieve provides automatic file recovery after a

system crash, so accidents and power failures don't turn into database disasters. Your Btrieve data always comes back intact.

Fully-relational data management. Using SoftCraft's entire family of products gives you a complete, fully-relational database management system. Rtrieve™ adds report writing capabilities for generating the reports you need. Xtrieve™ speeds users through database queries with interactive, on-screen menus—no command language or special syntax.

For professional programmers. Btrieve is the fast, reliable answer for all your application development. In any development language—BASIC, Pascal, COBOL, C, FORTRAN, and APL. With multikey access to records. Unlimited records per file. Duplicate, modifiable, and segmented keys. With Btrieve, you can develop better applications faster. And know they'll run—*fast*.

Now
available
for
XENIX

Suggested retail prices: Btrieve, \$245; Btrieve/N, \$595; Xtrieve, \$195; Xtrieve/N, \$395; Rtrieve, \$85; Rtrieve/N, \$175. Requires PC-DOS or MS-DOS 1.X, 2.X, or 3.X.

Btrieve, Xtrieve, and Rtrieve are trademarks of SoftCraft Inc.



SoftCraft Inc.

P. O. Box 9802 #917
Austin, Texas 78766
(512) 346-8380 Telex 358 200

COMMAND.COM needs about 18K and a programmer's editor may need 100K or more. We haven't even started to include a debugger, assembler, other libraries, object modules, or header files. In general, it would be nice to know the amount of space needed by the compiler, libraries, linker, and other assorted software the compiler needs to run.

2. The object-module format of the compilers was not mentioned. Micro-soft compatibility may be an important issue for some people, especially those who buy libraries.

3. The assembler that comes with the Mark Williams compiler does not use Intel mnemonics.

4. The review mentions that the large model for Lattice allows disabling of pointer range checking, which avoids a wraparound on a 64K boundary when doing pointer arithmetic. Nowhere was it mentioned how perilous such an undertaking can be on a project of any size!

5. Several of the compilers support in-line assembly, which is extremely useful. I did not see any mention of the DeSmet or DRI compilers sup-

porting that feature.

6. The DeSmet debugger, an excellent tool, wasn't even looked at!

7. The DeSmet compiler doesn't support the large memory model but does come with an overlay linker.

8. Although it can be useful to look at the size of the .EXE file produced by a compiler and linker, sometimes you want to know the sizes of the code and data produced by the compiler as well. Will the compiler tell you those statistics?

9. The reviewers didn't mention tradeoffs between compile/link time vs. speed of executable code (using an optimizer pass, for example).

10. The fact that Lattice and Computer Innovations automatically provide run-time stack-overflow checking was not mentioned.

I would like to have seen more of an attempt to separate out evaluations of the compilers from the associated libraries. I couldn't care less about the performance of libraries, since for applications that require speed and efficiency I write my own routines.

I have a second reason for writing. It seems to me that people deciding which compiler to choose might benefit from feedback from programmers actually using C compilers. Since C compilers are used in a wide variety of applications and environments under MS DOS, readers would likely get different kinds of information than a reviewer would provide. Why not allocate space for a running column where readers can sound off about their favorite compiler and give reasons why they like it?

If you readers like that idea, write to DDJ about the compiler you like to work with, mentioning aspects that haven't been covered. *[We would welcome this input.—Ed.]* Include your work environment (network, single-user, etc.); machine; any idiosyncracies or work requirements on programming style; and why the compiler you work with is the greatest thing since DDJ.

Tom Hogan
12801 Frost Rd.
Kansas City, MO 64138

DDJ

Open the door to diskette compatibility with UniForm!

Just \$69.95 turns one of your floppy drives into a "look-alike" of most popular CP/M* computers.

UniForm is the easy and inexpensive way to enable your CP/M computer to *directly* read, write and format diskettes from the most popular CP/M micros. This innovative program actually emulates the desired foreign drive. Files that were previously incompatible work flawlessly with your own software tools, such as word processors, database managers and spreadsheets. In addition, you can transfer files to or from MS-DOS.**

Menu-driven UniForm is mastered in just minutes—simply load, select the proper diskette format and go! Operation of your software is unchanged. At just \$69.95, compatibility never cost so little!



UniForm is available from your local computer dealer or Micro Solutions.

For users of IBM PC, PC-XT, PC-AT or compatibles, UniForm-PC provides direct access to CP/M data diskettes. Also just \$69.95.

Trademarks:
*Digital Research
**Microsoft Corporation

UniForm

MicroSolutions

125 South Fourth St.
DeKalb, IL 60115
815-756-3411

Mark Williams

Now the biggest name
in C compilers comes in a size
everybody can afford.
Let's C.™

Introducing Mark Williams' \$75 C compiler. Want to explore C programming for the first time? Or just on your own time? Now you can do it in a big way without spending that way. With Let's C.

This is no little beginner's model. Let's C is a powerful programming tool, packed with all the essentials of the famous Mark Williams C Programming System. The one chosen by Intel, DEC, Wang and thousands of professional programmers. The one that wins the benchmarks and the reviewers' praise:

"(This compiler) has the most professional feel of any package we tested..."—BYTE
"Of all the compilers reviewed, (it) would be my first choice for product development."—David W. Smith, PC WORLD

And now for more big news. Get our revolutionary csd C Source

Debugger for just \$75, too. You can breeze through debugging at the C source level ignoring clunky assembler code.

Affordable, powerful, debuggable. Mark Williams Let's C is the big name C compiler at a price you can handle. Get your hands on it now.

Mark Williams Let's C

- For the IBM-PC and MS-DOS
- Fast compact code plus register variables
- Full Kernighan & Ritchie C and extensions
- Full UNIX™ compatibility and complete libraries
- Small memory model
- Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc
- MicroEMACS full screen editor with source
- Supported by dozens of third party libraries
- Upgradeable to C Programming System for large scale applications development

Let's C Benchmark Done on an IBM-PC/XT, no 8087.
Program: Floating Point
from BYTE, August, 1983.

Exec Time in Seconds

Let's C	134.20
MS 3.0	347.45

Use this coupon or charge by calling toll-free:
1-800-MWC-1700. In Ill. call 312-472-6659.

Mark Williams Let's C

\$75

Please send me:

_____ copies of Let's C and _____ copies of csd (C Source Debugger)
at \$75 each. (Ill. residents add 7% sales tax.)

☐ Check ☐ Money Order ☐ Visa, MasterCard or
American Express

Name _____

Address _____

City _____ State _____ Zip _____

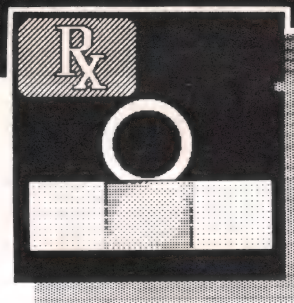
Card # _____ Exp. Date _____

Signature _____



**Mark
Williams
Company**

1430 West Wrightwood
Chicago, Illinois 60614



by D. E. Cortesi

Clinic to Close

The Clinic opened for business in May 1981. Now, four and one half years later, career changes compel me to close it down. The landlord will soon install new and interesting enterprises on the premises, but I thought I'd use our last meeting to look back and say thanks.

Two things come to mind when I reflect on the column. The first is trivial: that damned editorial We. I can't say why I hobbled myself with that foolish affectation in the first column nor why I didn't drop it in the second, but I did and then I didn't and from then on I felt compelled to continue an established style.

The strongest impression I retain is of the quality of the *DDJ* readership. As originally conceived, this column was to be written mostly by its readers with the Intern editing and supplying continuity, like a radio talk show in slow motion. It never quite worked out; you readers never provided enough original material to make a column.

But you responded when your interest was caught—my goodness how you responded! I posed Don Taylor's highly abstruse problem in the mathematics of graphics and a dozen of you sent detailed, multipage discussions heavy with matrix notation. I put up David Ross's BASIC program and 20 of you sent line-by-line discussions of what it did and how to do it better. Although the column was never written by readers in the way originally planned, you did supply a lot of material, correct a lot of my mistakes, and lend me your enthusiasm over the years.

Many of those who wrote were made into co-columnists and quoted in the various issues by name. They are named once more, with the issues

they graced by their presence, in the following list. For every one of those mentioned there were often two, three, or a dozen more who had the same insight but wrote later or in a less quotable way. My personal thanks to all who wrote—it's been a privilege to associate with such a classy bunch. You folks keep up the good work, you hear?

Allison, Dennis	8/81, 10/81	Dugan, Dan	4/82
Amelang, Loren	2/83	Edgar, Gerald	7/81
Asay, Matt	10/81	English, Glenn	11/85
Ashley, Allen	7/81, 10/81	Enterline, Tom	8/84
Backus, Rex	4/85	Falk, Richard	3/84
Bacon, Phillip	7/81	Farmer, Wayne	8/81, 10/81
Baenziger, Peter	8/84	Fay, Jack	11/85
Barker, L.	7/81, 10/81, 11/81, 10/82, 10/83	Feign, David	8/84
Barker, David	11/85	Floyd, Chet	9/84, 2/85
Barr, Michael	5/84, 8/85	Fordham, Malcom	4/84
Bates, Dan	9/81	Fritz, Randolph	5/83, 6/83
Bécus, Georges A.	9/81	Gardner, Calvin L.	8/84
Behler, Alan	5/83, 11/83	Glassey, C. Roger	12/81
Bomberger, Alan	10/83	Goldman, Oscar,	5/84
Brickell, Gavin	10/82	Goodman, Bob	7/81
Briggs, R. C.	8/84	Grant, Steve	8/84
Brown, Russel	8/81	Greenlaw, Richard	10/81
Butler, Edmund	7/81	Gunn, John Paul	2/85
Cage, Thomas W.	2/85, 6/85	Hahn, Harvey	5/83
Caldwell, Taylor	8/81	Hammond, Nick	10/82, 6/83, 10/83, 10/85
Canniff, Paul	4/85	Head, Gene	12/81
Carr, Thomas E.	9/81	Hills, Terence	11/81
Carter, Everett	8/85	Hoats, David L.	12/81
Castle, Pat	2/85	Hoffman, Lee H.	2/85
Chrapkiewicz, Thomas	8/84	Hole, Dr. William T.	2/85
Clark, David	2/84	Hoolko, Bob	7/81
Crowfoot, Norman	9/81	Howell, Edgar	4/82
Dale, Gary	1/85	Howell, Jim	5/82, 2/84, 4/85
Davis, Charles	5/85	Hutchinson, Aubrey	2/83-6/83
Davis, M. Patton	10/81	Hutchinson, W. G.	4/82
Dayhoff, Edward S.	11/81	Irving, Dwight	11/83
Dempsey, J.	3/83	Jackson, Terry	2/85
Dinsmore, Perry	11/84	Jensen, John Thayer	6/83
Dodd, Michael S.	11/85	Johnson, Bruce R.	2/85
Driscoll, Michael	9/81	Johnston, Bill	12/81
		Jones, James	4/85
		Kellogg, David	6/83
		Kendall, Bruce	8/82
		King, David	2/84
		Knipp, Ernest	10/82, 2/83
		Koehler, W. B.	11/83
		Komusin, Bruce	4/82, 5/82
		Lane, Tom	7/82
		Laurino, James	1/82, 7/83
		Leonard, M. David	7/81
		Levy, David	7/82

Lurie, Robert	10/81
McDaniel, C. T.	7/81
McDermott, Joseph	3/84
McKeon, Brian	4/85
McLanahan, David	1/84, 4/84, 8/84
McWorter, William A.	9/81, 5/83
Manay, Martin	10/81
Manns, Larry	11/85
Marchand, Fred	1/85
Marshall, Charles	3/84
Martin, Charles	2/85
Martin, John H.	8/81
Martin, Robert	10/81
Medlock, C. W.	5/84
Mesirov, Dick	3/84
Metzenthen, Bill	8/84
Meyer, Mike	10/84
Montgomery, Owen	5/84
Moore, Brian	3/83
Morgan, John	10/81
Neuhaus, Prof. H. J.	8/83
Newberry, Steve	2/84
Norling, Richard D.	11/84
Northrup, Mark	11/81
Novosielski, Gary P.	11/81
Nowell, Scott	11/81
Oakland, Richard L.	11/85
Oster, David	4/83
Palmer, John	10/82
Pasky, Robert	8/82
Pearson, Peter K.	8/85
Pierce, John	7/81
Pirko, Robert	3/83, 6/83, 7/83
Poli, Ray	4/82
Prince, Dr. Timothy C.	2/85
Pritchard, Steve	9/81
Pulver, Dusty	5/82
Purdum, Jack	2/84
Rivers, Wayne	1/85
Roberts, Glenn F.	7/84, 8/84
Rockwell, Charles	10/81
Ross, David	6/85, 8/85
Rouman, Jay S.	11/85
Rutiser, William A.	8/84
Saal, Harry	7/81
Sabin, Joseph	3/84, 2/85
Sackheim, Andrew	4/82
Safier, Orin	11/85
Salemma, A.	1/85
Samuel, Dan	4/82
Sand, Paul	1/85
Scardelis, Jim	4/84
Schafer, Dale	7/81
Schemm, John B.	2/85
Schuetz, Chuck	4/85
Schwab, Leonard	9/84
Semos, Steve	11/81

Shapin, Ted	4/84
Shaw, David A.	1/82, 4/82
Shaw, Jim	5/82
Shochat, David	11/85
Shuman, Dave	11/85
Smith, Billy	11/81
Smith, Burks	10/82, 2/83
Smith, Roger	8/85
Snow, J. R.	8/81
Stanek, Mitchell	9/81
Sullivan, Dave	4/84
Swearingen, Roger V.	2/85
Taylor, Bob	6/83
Taylor, Don	4/82, 8/82, 5/83
Thier, U.	7/84

Tilton, David S.	8/84, 10/84, 1/85, 4/85
Throckmorton, Jerry	10/81
Trout, Dana	10/82
Wagner, R. C.	7/84
Watzman, Barry	8/85
Wendel, J. G.	8/81
Wilkinson, Bill	7/82
Wilson, Ed	7/81
Wochnik, Michael	8/81
Wright, Don	9/82
Vieau, Dean	9/81

DDJ

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service No. 190.

DISCOVER THE LANGUAGE OF ARTIFICIAL INTELLIGENCE PROLOG V

At last! A Prolog with enough muscle to handle real-world applications for UNDER \$100! Discover why Japan has chosen Prolog as the vehicle for their "Fifth Generation Machine" project to design intelligent computers.

CHOOSE FROM TWO GREAT VERSIONS:

PROLOG V-Plus
\$9995

- ☐ More Than 100 Predefined Predicates
- ☐ Large Memory Model (to 640K)
- ☐ Floating Point Arithmetic
- ☐ 150-Page User's Manual and Tutorial plus Advanced Programming Documentation
- ☐ Co-Resident Program Editor
- ☐ Calls to Co-Resident Programs
- ☐ Text and Graphic Screen Manipulation

PROLOG V
\$6995

- ☐ 70 Predefined Predicates
- ☐ Small Memory Model
- ☐ Integer Arithmetic
- ☐ 122-Page User's Manual and Tutorial

UNBELIEVABLE UPGRADE POLICY
Tough decision? Buy PROLOG V and upgrade to PROLOG V-Plus within 60 days for only the difference in price plus a handling charge.

STANDARD FEATURES ON BOTH:

- ☐ Clocksin & Mellish-Standard Edinburgh Syntax
- ☐ Extensive Interactive Debugging Facilities
- ☐ Dynamic Memory Management (garbage collection)
- ☐ Custom-Designed Binder and Slipcase

THE CHOICE OF UNIVERSITIES

Generous university site licenses and an excellent teaching tutorial and reference guide have made PROLOG V the choice of universities nationwide. Call for details.

☐ **PAYMENT ENCLOSED** \$ _____
CA residents add 6% sales tax

☐ **CHARGE MY:** ☐ MasterCard ☐ Visa

Card No. _____ Exp. Date _____

Signature _____

Mr./Mrs./Ms. _____
(please print full name)

Address _____

City/State/Zip _____

PROLOG V-Plus \$99.95
PROLOG V 69.95
UPGRADE ONLY 40.00
Return factory diskette and \$30 plus \$10 Handling

SHIPPING:
\$ 5.00 U.S.
7.50 Canada
10.00 Caribbean,
Hawaii Air
20.00 Overseas Air

COD Orders Not Accepted
15 day check clearance

Not Copy
Protected



Interpreter for
MS-DOS/PC-DOS

PHONE ORDERS
(619) 483-8513

NO RISK OFFER
Examine the documentation at our risk for 30 days. If not fully satisfied, return with disk still sealed for full refund.



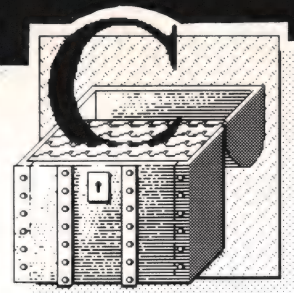
**CHALCEDONY
SOFTWARE**

5580 LA JOLLA BLVD.
SUITE 126 D
LA JOLLA, CA
92037

Circle no. 178 on reader service card.

A New Shell for MS DOS, Part 1: IBM Cursor Control and an Fgets that Edits

by Allen Holub



Back in March we presented a new root module for the CP/M-80 version of the Aztec C compiler. Since then, several people have written to say that the proper place for wildcard expansion, pipes, redirection, and the like is in the shell (i.e., the command interpreter), not in the C root module. There's no point in bloating the size of your programs when you don't have to. I sympathize with this attitude, but frankly, I don't want to write a new CCP for CP/M. Digital Research just makes this process too hard.

MS DOS, however, is another matter. The command interpreter, `command.com`, is just a program and can be replaced with any other program. Moreover, the DOS interface library provides much more in the way of system level functions than does CP/M. Most important, DOS Version 2 and higher supports an `exec` function that allows one program to load and execute a second program.

Because most of my gripes about MS DOS are really aimed at the behavior of `command.com`, it seemed to me that writing a new shell for MS DOS would be more productive than complaining for the rest of my life about the features I didn't have. C is, after all, a language remarkably well suited for writing operating systems. So, over the next few months we will look at a new shell for MS DOS.

I've implemented most of what I like about the the Unix C and Bourne shells: command line wild card expansion, redirection, history, shell variables, a reasonable language for writing shell scripts (.bat files under MS DOS) and the lot. I've also taken the opportunity to add a couple of features that I've often wished were present in the Unix shells, such as an interactive command line editor.

String Routines

We'll start looking at the shell itself next month. This month we're going to look at several support routines. As usual, these are all designed to be useful in their own right. They make up five sets. The first three listings are routines for string manipulation. They are small, but useful. `Next()` (Listing One page 19) gets from a string the next object separated by a delimiter. Its calling syntax is:

```
char *next( linep, delim, esc )
char **linep;
int delim, esc;
```

`Linep` is the address of a pointer that points at the first character of the string to be processed. `Delim` is the delimiter that separates objects. `Esc` is an escape character. The routine skips leading white-space (a space or tab) and remembers the position of the first non-white character. It then looks for a delimiter not preceded by an escape character and replaces the delimiter with a '\0.' It returns a pointer to the remembered non-white character and it modifies `*linep` to point to the first character beyond where the delimiter used to be. It won't modify `*linep` to point past the end of the string. If `delim` is a white character, leading white space won't be skipped.

Listing Two (page 19) and Listing Three (page 19) are two routines for copying. `Cpy(dest,src)` is functionally the same as `strcpy()`. It copies the source string (`src`) to the destination string (`dest`). There is, however, one difference: whereas `strcpy` returns its first argument, `cpy` returns a pointer to the NULL at the end of the destination string. This allows us to add onto the destination string without first having to scan through the string

to find its end. The second copy routine is `cptolower(dest,src)`. `Cptolower` works just like `cpy()`, except all uppercase characters are mapped to lower case as they are copied.

Moving the Cursor and Writing Characters

Listing Four (page 20) is a collection of low-level routines for cursor manipulation and character output for the IBM PC. These routines use direct ROM BIOS calls, and so are much faster than the usual output routines, which use the normal DOS functions¹. Moreover, they let you move the cursor around without having to install the `ANSI.SYS` driver, which is excruciatingly slow. IBM has been very good about maintaining the ROM BIOS interface over different versions of the operating system. As long as you access the BIOS via the correct interrupt mechanism, the routines are portable.

I've used the `int86()` function from Lattice to generate the system video interrupt. Microsoft C has an identical function, so these routines will port to Microsoft C without any problems. The `dos()` function printed in this column in July 1985 can also be modified quite easily to generate a video interrupt. Just replace the `INT 21H` instruction on line 68 with an `INT 10H` instruction.

Listing Four contains the following routines:

`getpage()`

This routine gets the currently active video page. When you are in text mode, several pages are available for writing, though only one is displayed at any given time. `Getpage()` returns the number of the page now being displayed. This information is needed by some of the other routines.

cursize(top, bot)

This routine adjusts the cursor size. Every character is represented on the display by a fixed number of scan lines. Cursize() causes the cursor to extend from the top scan line indicated to the bottom line indicated. For example, the standard monochrome display uses 14 lines numbered 0-13, the color adapter uses 8, numbered 0-7. So, on the color adapter, a normal underline cursor can be set up with the call cursize(6,7). You can make a large block cursor with the call cursize(0,7). Cursize(7,0) creates a two-part cursor, one line over the letter and a second line under the letter. Cursize(8,8) will make the cursor disappear.

posn = gcur(pagenum);

int pagenum;

short posn;

This routine returns the position of the cursor on the indicated video page. The position is returned in a single 16 bit short; the row number is in the top byte and the column number is in the bottom byte.

scur(posn, page)

short posn;

This modifies the cursor to rest at posn, which may be a value returned from gcur().

posn = getcur();

setcur(posn);

These routines work just like scur() and gcur() except that they access the page currently being displayed.

wchar(c)

int c;

This routine writes a single character to the screen. It is dramatically faster than putchar()—at least the one that comes with the Lattice compiler. Wchar() moves the cursor as it prints, just like putchar(). However, the only control codes it recognizes are carriage return (\r), line feed (\n), bell (\007), and backspace (\b), so don't expect it to expand tabs. Also, \n is not interpreted as a "newline," but as a line feed. That is, it will get you to the current column on the next line. Use \r to get to the left edge of the current line.

wstr(s, move)

char *s;

int move;

This routine prints an entire string to the screen. If move is true, the cursor ends up positioned just past the end of the printed string. Otherwise, the cursor remains over the first character of the string. Wstr() uses wchar() as its output function.

An Fgets() that Edits

Listing Five (page 20) is an editing input function. Let me point out at the start that this function is general purpose in nature, even though it's written specifically for an IBM PC. It's a relatively simple matter to modify the routines for any terminal that has an addressable cursor. The primary access routine in this module is

char *efgets(buf, maxline, fp)

char *buf;

int maxline;

FILE *fp;

This is functionally similar to

fgets(), though there are several major differences. First, a pointer to the end (instead of to the start) of the loaded buffer is returned on success. A NULL is still returned on end of file. Second, line continuation is supported. If a line ends with a backslash (\), the backslash is deleted and the line is concatenated with the next line. The major differences from fgets() are apparent when fp is set to stdin. In this case several interactive editing functions are supported:

- **LEFT-CURSOR** (hit the left cursor key) moves the cursor to the left without erasing anything.
- **RIGHT-CURSOR** moves the cursor to the right.
- **^LEFT-CURSOR** (hold down the CTRL key and hit the left cursor key) positions the cursor at the beginning of the previous word.
- **^RIGHT-CURSOR** positions the cursor at the beginning of the next word.
- **HOME** positions the cursor at home position on the current line. Home is

MS-DOS, UNIX, APPLE MAC, CP/M, NETWORKS and MORE. ONE c-tree ISAM DOES THEM ALL!

The creator of Access Manager™ brings you the most powerful C source code, B+ Tree file handler: **c-tree™**

- multi-key ISAM and low-level B+ Tree routines
- complete C source code written to K&R standards
- single-user, network and multi-tasking capabilities
- fixed and variable record length data files
- virtually opened files accommodate limited file descriptors
- no royalties on application programs

\$395 COMPLETE

Specify diskette format:

- 5¼" MS-DOS
- 8" CP/M
- 3½" Mac
- 8" RT-II



For VISA, MC and COD orders
call (314) 445-6833
FairCom
2606 Johnson Drive
Columbia, MO 65203

© 1985 FairCom

The following are trademarks: c-tree and the circular disk logo—FairCom; MS—Microsoft Inc.; CP/M and Access Manager—Digital Research Inc.; Unix—AT&T; Apple—Apple Computer Co.

Circle no. 93 on reader service card.

defined as the position of the cursor when the routine was entered. For example, if the cursor was in column 5 when `efgets()` was called, HOME puts it back in column 5. In fact, in this example there's no way to get the cursor into columns 1-4 from `efgets`. If a line is continued with a `\<CR>` combination, the cursor is moved to the HOME position of the next line, not to the left edge of the screen.

- END positions the cursor just after the rightmost character on the line.
- ESC is for an abort. The buffer is cleared, but the characters on the screen are not erased. The routine re-

turns immediately. -1 is returned when an ESC is encountered.

- DEL deletes the character on which the cursor is resting and closes up the rest of the line to fill the gap. (i.e., if you delete the X in the string "aaaaXoooo," you end up with "aaaaoooo" on the screen, rather than "aaaa oooo").
- ^H is a destructive backspace. It moves the cursor left one space and then deletes that character, closing up the line to fill the gap.
- ^X erases the entire line and clears the buffer. However, `efgets()` doesn't return.

- ^M (CR) or Enter (LF) positions the cursor at the left edge (not the home position) of the next line and cause `efgets()` to return to the caller. A pointer to the end of the buffer (to the \0) is returned on success, NULL on end of file.

Typing any printing character causes that character to be printed at the current cursor position and the cursor to move right one space. Typing anything else has no effect. The cursor is never allowed to go past the end of the buffer, as specified by the parameter `maxline` in the `efgets()` call. However, when in editing mode, the cursor is not allowed to go past the end of the current line, even if `maxline` is longer (the bell will ring if you try). In this case, you can get to the next line with a `\<CR>`, but you can't edit anything on the previous line.

Conclusion

So that's the beginning. Next month we'll add some more routines to the pile and incorporate them into a simple MS DOS shell. In the following month we'll add various capabilities to that shell.

[IBM PC readable versions of the listings for the entire shell and for MS DOS versions of various Unix utilities (e.g., grep, ls) will be made available through DDJ in the next couple of months. Watch this column for more details.—ed]

Notes

- ¹ An excellent description of the IBM BIOS routines and how to use them is in *The Peter Norton Programmer's Guide to the IBM PC*, by Peter Norton (Bellevue: Microsoft Press: 1985).

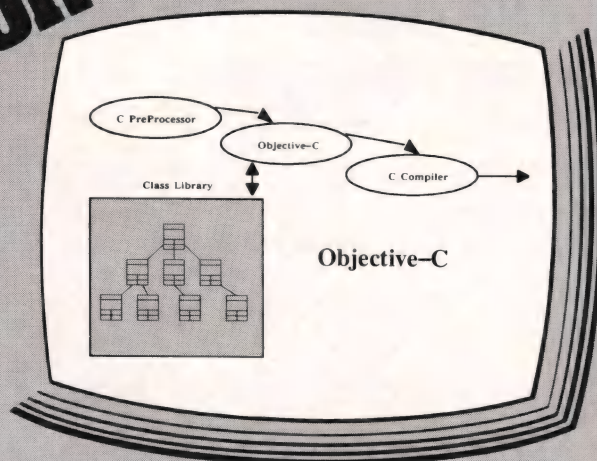
DDJ

(Listing begins at right)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 191.

NEW! Object-Oriented Programming on a PC!



Objective-C™ is an object-oriented programming language and a fully documented library of reusable components...adding messages, objects and inheritance to C language. Applications written in Objective-C are fully compatible with other Objective-C compilers running under UNIX, VMS or AOS.

Objective-C provides the productivity of object-oriented programming, while

retaining the portability and efficiency of C. PPI also provides comprehensive technology transfer to insure that your programmers fully understand this exciting new technology.

PPI's Objective-C compiler generates C code, which requires the Microsoft V3 C compiler running under MS/DOS.

At \$500 Objective-C is affordable. Order today!

PRODUCTIVITY PRODUCTS INTERNATIONAL

27 Glen Road, Sandy Hook, CT 06482. (203) 426-1875.



Listing One

```
1: /*      NEXT.C:      Skip to the next delimiter seperated object
2: *
3: */
4:
5: #define iswhite(c) ( (c) == ' ' || (c) == '\t' || (c) == '\n' )
6:
7:
8: char      *next( linep, delim, esc )
9: char      **linep;
10: {
11:     /* Linep is the address of a character pointer that points to
12:      * a string containing a series of delim seperated objects.
13:      * Next will return a pointer to the first non-white object in
14:      * *linep, replace the first delimiter it finds with a null, and
15:      * advance *linep to point past the null (provided that it's not
16:      * at end of string). 0 is returned when an empty string is passed
17:      * to next(). White space may be used as a delimiter but
18:      * in this case white space won't be skipped. A delimiter preceded
19:      * by "esc" is ignored. Quoted strings are copied verbatim.
20:     */
21:
22:     register char      *start, *end ;
23:     int                 inquote = 0;
24:
25:     if( !**linep )
26:         return 0;
27:
28:     start = *linep;
29:
30:     if( !iswhite(delim) )
31:         for( ; iswhite(*start) ; start++ )
32:             ;
33:
34:     for( end = start; *end && (*end != delim || inquote) ; end++ )
35:     {
36:         if( *end == esc && *(end+1) )
37:             end++;
38:
39:         else if( *end == '"' || *end == '\'' )
40:             inquote = ~inquote;
41:     }
42:
43:     if( *end )
44:         *end++ = '\0';
45:
46:     *linep = end;
47:     return start;
48: }
```

End Listing One

Listing Two

```
1: char      cptolower( dest, src )
2: char      *dest, *src;
3: {
4:     /*      Copy src to dest, mapping all upper case letters to lower
5:     *      case. src and dest may be the same.
6:     */
7:
8:     for( ; *src ; src++ )
9:         *dest++ = ('A' <= *src && *src <= 'Z')
10:             ? *src + ('a'-'A') : *src ;
11:
12:     *dest = '\0' ;
13:     return dest ;
14: }
```

End Listing Two

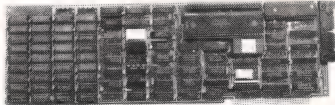
Listing Three

```
1: char      *cpy( dest, src )
2: char      *dest, *src;
3: {
4:     /*      Works like strcpy but returns a pointer to the new end
5:     *      of string (ie. to the null).
6:     */
7:
8:     while( *src )
9:         *dest++ = *src++ ;
10:
11:     *dest = 0;
12:     return dest;
13: }
```

End Listing Three

(Listing Four begins on next page)

68000 CO-PROCESSING For IBM PC, PC/XT and COMPATIBLE SYSTEMS



Now you can add the MOTOROLA 68000 16/32 Bit Processor to your PC via use of the Pro 68 Advanced Technology Co-Processor. Enjoy all of the performance benefits of the 68000 processor without sacrificing your current PC system. Consider these impressive standard features of Pro 68:

- High Speed MOTOROLA 68000 micro processor
- 10Mhz no wait state design (3 times faster than the IBM PC/AT)
- True 16/32 bit technology
- For use on IBM PC, PC/XT or compatible systems
- On board 16 bit parity checked memory, 256K to 1024K
- Two serial I/O ports for multi user interface
- Provisions for the high speed NS32081 math processor
- High speed proprietary dual port host bus interface
- Parallel or array processing via multi processor architecture
- MS/PC DOS RAM disk driver program
- Choice of two popular integrated 16/32 bit operating systems:
 - CPM68K from Digital Research Inc.
 - Full suite of development tools
 - "C" compiler with floats and UNIX I/O library
 - Many third party compatible languages and applications
 - OS9/6800 from MICROWARE Corporation
 - UNIX look alike with multi user/ multi tasking, shell, hierarchical disk directory, record and file lock, pipes and filters
 - Full suite of development tools
 - UNIX V compatible "C" compiler
 - Optional languages include BASIC, ISO PASCAL, FORTRAN 77.

Pricing from \$1195 includes Pro68 with 256K, OS, and MS/PC DOS RAM disk driver. HSC also manufactures and markets a full line of co-processors and RAM disks for use on Z80 based systems.

DISTRIBUTORS:

Australia-Computer Transition Systems
...03-537-2768
Great Britain-System Science
...01-248-062
West Germany-DSC International
...089-723-1125
Canada Remote Systems
...416-239-2835

Dealer, Distributor and OEM inquiries invited.

Hallock Systems Co., Inc.

267 North Main Street
Herkimer, NY 13350
(315) 866-7125



C Chest (Listing continued, text begins on page 16) Listing Four

```

1: #include <stdio.h>
2: #include "/lc/dos.h"
3:
4: /*  VBIOS.C          Various cursor and i/o routine using
5: *                      the bios interrupts (see below for greater detail):
6: *
7: *                      Copyright (C) 1985 Allen I. Holub. All rights reserved.
8: *
9: *                      Externally accessible routines:
10: *
11: *      int      getpage ( )           Get active video page #
12: *      void     cursize (top,bot)     Set cursor size
13: *      void     scur  (posn, page) short posn; Set cursor position
14: *      short    gcur  (pagenum)      Get cursor position
15: *      void     setcur (posn) short posn; Set cur pos on current page
16: *      short    getcur ( )           Get cur pos from cur page
17: *      void     wchar  (c)           write a single character
18: *      wstr     wstr   (s, move) char *s; write a string
19: *
20: */
21:
22: #define VIDEO_INT    0x10 /*      Video interrupt      */
23:
24: #define CUR_SIZE     0x1 /*      Set cursor size      */
25: #define SET_POSN     0x2 /*      Modify cursor posn   */
26: #define READ_POSN    0x3 /*      Read current cursor posn */
27: #define WRITE_TTY    0xe /*      Write character & move cursor */
28: #define GET_VMODE    0xf /*      Get current video mode & disp pg */
29:
30: /*-----*/
31:
32: static union REGS      Regs;
33:
34: /*-----*/
35:
36: int      getpage ( )
37: {
38:     /*      Returns the currently active display page number
39:     */
40:
41:     Regs.h.ah = GET_VMODE;
42:     int86 ( VIDEO_INT, &Regs, &Regs );
43:
44:     return (int) Regs.h.bh ;
45: }
46:
47: /*-----*/
48:
49: cursize ( top_line, bot_line )
50: {
51:     /*      Scan lines are numbered 0 at the top and 7 at the bottom.
52:     *      if the two are reversed you'll get a 2 part cursor.
53:     *      Top_line determines the position of the top scan line
54:     *      of the cursor, bot_line is the bottom. A normal cursor
55:     *      can be created with cursize(6,7). Cursize(0,7) will
56:     *      fill the entire area occupied by a character. Cursize(0,1
57:     *      will put a line over the character rather than under it.
58:     */
59:
60:     Regs.h.ch = top_line ;
61:     Regs.h.cl = bot_line ;
62:     Regs.h.ah = CUR_SIZE ;
63:     int86 ( VIDEO_INT, &Regs, &Regs );
64: }
65:
66: /*-----*/
67:
68: scur ( posn, pagenum )
69: short    posn;
70: {
71:     /*      Modify current cursor position. The top byte of "posn"
72:     *      value holds the row, the bottom by the column.
73:     *      Pagenum is the video page number.
74:     */
75:
76:     Regs.x.dx = posn ;
77:     Regs.h.bh = pagenum ;
78:     Regs.h.ah = SET_POSN ;
79:     int86 ( VIDEO_INT, &Regs, &Regs );
80: }
81:
82: short    gcur ( pagenum )
83: {
84:     /*      Get current cursor position. The top byte of the return
85:     *      value holds the row, the bottom by the column.
86:     *      Pagenum is the video page number.
87:     */
88:
89:     Regs.h.bh = pagenum ;
90:     Regs.h.ah = READ_POSN ;
91:     int86 ( VIDEO_INT, &Regs, &Regs );
92:
93:     return ( Regs.x.dx );

```



```

94: }
95:
96: /*      Setcur and getcur work just like scur and gcur except that they
97: *      access the current video page.
98: */
99:
100: setcur( posn )
101: short   posn;
102: {
103:     scur( posn, getpage() );
104: }
105: short   getcur()
106: {
107:     return gcur( getpage() );
108: }
109:
110: /*-----*/
111:
112: wchar( c )
113: {
114:     /*      Write a character to the screen in TTY mode. Only normal
115:     *      printing characters, BS, BEL, CR and LF are recognized.
116:     *      The cursor is automatically advanced and lines will wrap.
117:     */
118:
119:     Regs.h.bl = 0;          /* Use current color */
120:     Regs.h.al = c;
121:     Regs.h.ah = WRITE_TTY ;
122:     int86( VIDEO_INT, &Regs, &Regs );
123: }
124:
125: wstr( str, move_cur )
126: char   *str;
127: {
128:     /*      Write a string to the screen in TTY mode. If move_cur is
129:     *      true the cursor is left at the end of string. If not
130:     *      the cursor will be restored to its original position
131:     *      (before the write).
132:     */
133:
134:     register short   posn;
135:
136:     if( !move_cur )
137:         posn = getcur();
138:
139:     while( *str )
140:     {
141:         Regs.h.bl = 0 ;
142:         Regs.h.al = *str++ ;
143:         Regs.h.ah = WRITE_TTY ;
144:         int86( VIDEO_INT, &Regs, &Regs );
145:     }
146:
147:     if( !move_cur )
148:         setcur( posn );
149: }
150:
151: /*-----*/
152: #ifdef DEBUG
153:
154: main()
155: {
156:     cursize( 0, 7 );
157:     wstr( "The large cursor should be on the 'T'", 0 );
158:     getchar() ;
159:
160:     cursize( 6, 7 );
161: }
162:
163: #endif

```

End Listing Four

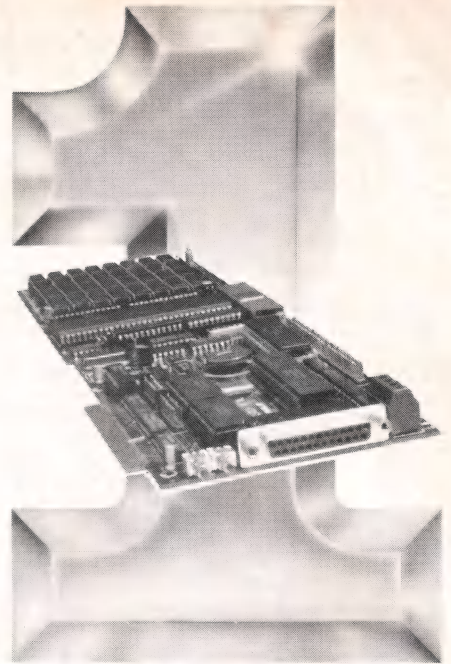
Listing Five

```

1: #include <stdio.h>
2:
3: /*      EFGETS.C      An editing version of efgets. Recognizes \<CR>
4: *      line termination and supports editing when input
5: *      is from stdin.
6: *
7: *      Copyright (C) 1985 Allen I. Holub. All rights reserved.
8: *
9: *-----
10: *
11: * Externally accessible routines:
12: *
13: * void ptail (bp, end, move)      Print string from bp to end. Move
14: *                                cursor to end if move is true.
15: * char *egets (start, bufsize)    Get a string from stdin w/ editing.
16: *                                get at most bufsize-1 chars.
17: * char *getl (buf, maxline, fp)    like fgets but returns pointer to
18: *                                end of input string on success.
19: * char *efgets (buf, maxline, fp) like getl but uses egets for
20: *                                standard input rather thangetc.

```

(Continued on next page)



**Number One
in Performance**
68010/68000
Coprocessor for
IBM/AT/XT/PC-
8/10/12.5mz No Wait States
\$1395⁰⁰ Qty. 1

FEATURES

- 1-2 MB RAM (1MB Standard)
- 16K-64K EPROM
- 2-8 Serial Ports
- Async/Sync/Bisync Communications
- Battery-backed Real Time Clock
- Battery-backed 2K-8K RAM
- 2 Parallel Ports
- Memory-mapped Dual-port BUS
- 3-9 Users Per Board (3 Standard)
- Up To 16 Boards Per AT/XT/PC
- Can Operate As Standalone Processor

SOFTWARE

- OS9 (Powerful UNIX-like Multi-user OS)
- CPM/68K
- Software selectable OS including concurrent PC DOS/OS-9 or CPM/68K operation
- Support Module for IBM Graphics
- High-speed Local/Global Disk Caching
- Basic, Pascal, Fortran, C, and COBOL

IBM is a registered trademark of International Business Machines Corporation.
OS/9 is a registered trademark of Microsoft Corporation.
CPM/68K is a registered trademark of Digital Research Corporation.
MS-DOS is a registered trademark of Microsoft Corporation.



4704 W. JENNIFER SUITE 105 • FRESNO, CA 93711
209/276-2345

Circle no. 173 on reader service card.

GROWING OLD?

...waiting
for C programs to
compile and link?



Use **C-terp**
the complete C interpreter
This is the product you've been
waiting (and waiting) for!

Increase your productivity and avoid agonizing waits. Get instant feedback of your C programs for debugging and rapid prototyping. Then use your compiler for what it does best...compiling efficient code ...slowly.

C-terp Features

- Full K&R C (no compromises)
- Complete built-in screen editor--no half-way house, this editor has everything you need such as multi-files, inter-file move and copy, etc. etc. For the ultimate in customization, editor source is available for a slight additional charge of \$98.00.
- Fast--Linking and semi-compilation are breath-takingly fast. (From edit to run completion in a fraction of a second for small programs.)
- Convenient--Compiling and running are only a key-stroke or two away. Errors direct you back to the editor with the cursor set to the trouble spot.
- Object Module Support-- Access functions and externals in object modules produced by your compiler. **New:** We are now supporting **Microsoft 3.0, Mark Williams & Aztec C** in addition to C.I. C86 & Lattice.
- Complete Multiple Module Support.
- Symbolic Debugging-- Set breakpoints, single-step, and directly execute C expressions.
- Many more features including batch mode and 8087 support.
- **Price: \$300.00 (Demo \$45.00) MC, VISA**
Price of demo includes documentation & shipping within U.S. PA residents add 6% sales tax. Specify compiler.
- C-terp runs on the IBM PC (or compatible) under DOS 2.x with a suggested minimum of 256kb of memory. It can use all the memory available.

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

*Trademarks: C86 (Computer Innovations), Lattice (Lattice Inc.), IBM (IBM Corp.), C-terp (Gimpel Software), Microsoft (Microsoft), Aztec (Marx)

C Chest (Listing continued, text begins on page 16) Listing Five

```

21: *
22: *      FILE *fp;
23: *      char *buf, *bp, *end;
24: *      int move, bufsize, maxline;
25: */
26:
27: extern int  getpage ();          /* Sources in vidbios.c */
28: extern void setcur ();          /*
29: extern short getcur ();         /*
30: extern void wchar ();           /*
31: extern int  dos ();             /* Part of the Lattice standard library */
32: extern void movemem ();        /*
33:
34: /*      Values returned from DOS when cursor keys etc. are hit:
35: */
36:
37: #define _LEFT      75
38: #define _RIGHT     77
39: #define _CTL_LEFT  115
40: #define _CTL_RIGHT 116
41: #define _INS       82
42: #define _DEL       83
43: #define _HOME      71
44: #define _END       79
45:
46: /* The above are mapped as follows by getkey
47: */
48:
49: #define LEFT      0x80
50: #define RIGHT     0x81
51: #define CTL_LEFT  0x82
52: #define CTL_RIGHT 0x83
53: #define INS       0x84
54: #define DEL       0x85
55: #define HOME      0x86
56: #define END       0x87
57:
58: #define BDOS_IN   8      /* raw (non echo) input function */
59:
60: #define CNTL_C    0x03    /* ^C */
61: #define CNTL_Z    0x1a    /* ^Z */
62: #define BEL       0x07    /* ^G */
63: #define ESC       0x1b    /* ^[ */
64: #define CAN       0x18    /* ^X */
65:
66:
67: /*-----*/
68:
69: static int  getkey()
70: {
71:     /*      Return a key from the keyboard. Keys are gotten in raw
72:     *      input mode and mapped as specified above if necessary.
73:     */
74:
75:     register int  c;
76:     static int  ateof = 0;
77:
78:     if( ateof )
79:         return EOF;
80:
81:     if( ! (c = bdos(BDOS_IN)) )      /* Special function key */
82:     {
83:         switch( bdos(BDOS_IN) )
84:         {
85:             case _LEFT:      return( LEFT );
86:             case _RIGHT:     return( RIGHT );
87:             case _CTL_LEFT:  return( CTL_LEFT );
88:             case _CTL_RIGHT: return( CTL_RIGHT );
89:             case _INS:       return( INS );
90:             case _DEL:       return( DEL );
91:             case _HOME:      return( HOME );
92:             case _END:       return( END );
93:             default:         return( NULL );
94:         }
95:     }
96:     else if( c == '\r' )              /* map ENTER key to '\n' */
97:     {
98:         return( '\n' );
99:     }
100:     else if( c == CNTL_C || c == CNTL_Z )
101:     {
102:         ateof = 1;
103:         return EOF;
104:     }
105:     else
106:         return c;
107: }
108:
109: /*-----*/
110:
111:
112: ptail( bp, end, move )

```

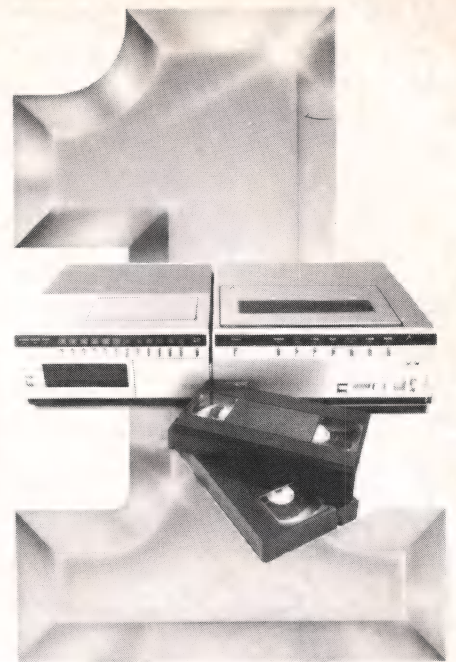


```

113: register char *bp, *end;
114: {
115:     /* Print out all characters between bp and end (inclusive)
116:      * without modifying the current cursor position. If move
117:      * is zero the cursor will not change position, otherwise
118:      * the cursor will be left pointing to the character
119:      * referenced by end.
120:      */
121:
122:     register short posn;
123:
124:     if( !move )
125:         posn = getcur();
126:
127:     for( ; *bp && bp <= end; wchar(*bp++) )
128:         ;
129:
130:     if( !move )
131:         setcur( posn ); /* put cursor under first character */
132:     else
133:         wchar('\b'); /* put cursor under last character */
134: }
135:
136: /*-----*/
137:
138: char *egets( start, bufsize )
139: char *start;
140: {
141:     /* Get a string with editing. If bufsize is wider than your
142:      * screen, strange things will happen when you try to use
143:      * the editing functions. If you access this function
144:      * via fgetline() then it will input longer lines in 78
145:      * character chunks and \<CR> can be used to extend a line.
146:      */
147:     *
148:     ^H or BACKSPACE    Destructive backspace, close up
149:                        remainder of string to fill hole.
150:     *
151:     LEFT CURSOR        Non-destructive backspace
152:     *
153:     RIGHT CURSOR       Move right one character.
154:     *
155:     ^LEFT CURSOR       Left to previous word or line start
156:     *
157:     ^RIGHT CURSOR      Right to next word or line end
158:     *
159:     HOME               Left edge of line
160:     *
161:     END                Right edge of line
162:     *
163:     CR or LF           Terminate line.
164:     *
165:     ^X                 Erase entire line but don't return.
166:     *
167:     ESC                Return a null string immediatly.
168:     *
169:     DEL                Delete a current cursor position and
170:                        close up to fill hole.
171:     *
172:     Any printing character Enter that char at cursor posn
173:     *
174:     Anything else      Ring the bell.
175:     *
176:     The bell will also ring if you try to move the cursor
177:     *
178:     past either the left or right edges of the buffer.
179:     *
180:     Return a pointer to the end of string normally, return 0
181:     *
182:     on EOF and return -1 when ESC is encountered.
183:     */
184:
185:     register char *bp; /* Points at current cursor position */
186:     register char *end; /* Points at largest possible cur pos. */
187:     register char *maxbp; /* Points at rightmost char on line */
188:     register int c; /* Current character. */
189:     short home; /* place to remember the leftmost cursor
190:                  * position.
191:                  */
192:
193:     end = start + (bufsize-1);
194:     *end-- = '\0';
195:     bp = start;
196:     maxbp = start;
197:
198:     /* Fill the entire buffer with spaces
199:     */
200:
201:     home = getcur(); /* Get the current cursor
202:                      */
203:                      /* position.
204:                      */
205:
206:     if( *bp ) /* If the buffer isnt empty */
207:     { /* print out its contents */
208:         while( *bp ) /* and set maxbp to point
209:                     */
210:             wchar( *bp++ ); /* at the previous end of
211:                             */
212:         maxbp = bp; /* string.
213:                     */
214:         setcur( home );
215:     }
216:
217:     for( ; bp <= end; *bp++ = ' ' ) /* and then fill the rest
218:                                     */
219:         ; /* of it with spaces.
220:         */
221:
222:     bp = start;
223:
224:     /* bp points into the bufer at the current cursor location.
225:     * end points at the righmost place that the cursor movement
226:     * commands can get us. There is actually one more place
227:     * in the buffer. Get the line:
228:     */
229:
230:     while( (c = getkey()) != '\n' && c != EOF )
231:     {
232:         switch( c )

```

(Continued on next page)



Number One In Performance

Hard Disk Intelligent VCR Backup for AT/XT/PC

FEATURES

- High speed microprocessor controlled backup (68008)
- Two channel interface
- Built in LAN channel
- Software control of most VCR functions including Fast Forward, Rewind, and auto backup using VCR timer capabilities
- Economical VHS or Beta formats



4704 W. JENNIFER, SUITE 105 • FRESNO, CA 93711
209/276-2345

Circle no. 174 on reader service card.

PRIME FEATURES

- Execute DOS level commands in HS/FORTH, or execute DOS and BIOS functions directly.
- Execute other programs under HS/FORTH supervision. (editors debuggers file managers etc)
- Use our editor or your own.
- Save environment any time as .COM or .EXE file.
- Eliminate headers, reclaim space without recompiling.
- Trace and decompile.
- Deferred definition, execution vectors, case, interrupt handlers.

HS FORTH

- Full 8087 high level support. Full range transcendentals (tan sin cos arctan logs exponentials)
- Data type conversion and I/O parse/format to 18 digits plus exponent.
- Complete Assembler for 8088, 80186, and 8087.
- String functions - (LEFT RIGHT MID LOC COMP XCHG JOIN)
- Graphics & Music
- Includes Forth-79 and Forth-83
- File and/or Screen interfaces
- Segment Management
- Full megabyte - programs or data
- Fully Optimized & Tested for: IBM-PC XT AT and JR COMPAQ and TANDY 1000 & 2000 (Runs on all true MSDOS compatibles!)
- Compare
BYTE Sieve Benchmark jan 83
HS/FORTH 47 sec BASIC 2000 sec
with AUTO-OPT 9 sec Assembler 5 sec
other Forths (mostly 64k) 55-140 sec
FASTEST FORTH SYSTEM AVAILABLE.

**TWICE AS FAST AS OTHER
FULL MEGABYTE FORTHS!**

(TEN TIMES FASTER WHEN USING AUTO-OPT!)

HS/FORTH, complete system only: \$270.



HARVARD SOFTWORKS

P.O. BOX 69
SPRINGBORO, OH 45066
(513) 748-0390

C Chest (Listing continued, text begins on page 16) Listing Five

```

211: {
212: case LEFT:          /* Non-destructive backspace */
213:
214:     if( bp > start )
215:     {
216:         wchar( '\b' );
217:         --bp;
218:         break;
219:     }
220:
221:     wchar( BEL );
222:     break;
223:
224: case '\b':          /* Destructive backspace */
225:
226:     if( bp <= start )
227:     {
228:         wchar( BEL );
229:         break;
230:     }
231:
232:     wchar( '\b' );
233:     --bp;
234:
235:     /* fall through to delete case */
236:
237:
238: case DEL:           /* Delete character and close up */
239:
240:     if( bp >= maxbp ) /* nothing to delete */
241:         break;
242:
243:     movmem( bp+1, bp, maxbp-bp );
244:     *maxbp = ' ';
245:     ptail( bp, maxbp-- , 0 );
246:     break;
247:
248: case CTL_LEFT:      /* Cursor to start of previous word */
249:
250:     if( bp > start )
251:     {
252:         do {
253:             --bp;
254:             wchar( '\b' );
255:
256:             } while( bp > start && *bp == ' ' );
257:
258:             while( bp > start && *bp != ' ' )
259:             {
260:                 --bp;
261:                 wchar( '\b' );
262:             }
263:
264:             if( *bp == ' ' )
265:                 wchar( *bp++ );
266:             break;
267:
268:         }
269:
270:         wchar( BEL );
271:         break;
272:
273: case HOME:          /* Cursor to left extreme */
274:
275:     bp = start ;
276:     setcur( home );
277:     break;
278:
279: case ESC:           /* Erase entire line and return */
280:
281:     *start = 0;
282:     wchar( '\r' );
283:     wchar( '\n' );
284:     return -1 ;
285:
286: case CAN:           /* Erase entire line */
287:
288:     setcur( home );
289:     for( bp = start; bp < maxbp; *bp++ = ' ' )
290:         wchar( ' ' );
291:
292:     setcur( home );
293:     maxbp = bp = start;
294:     break;
295:
296: case RIGHT:         /* Cursor right one character */
297:
298:     wchar( ( bp < end ) ? *bp++ : BEL );
299:     break;
300:
301: case CTL_RIGHT:     /* Advance to next word */
302:
303:     while( bp < maxbp && *bp != ' ' )
304:         wchar( *bp++ );
305:
306:     while( bp < maxbp && *bp == ' ' )

```

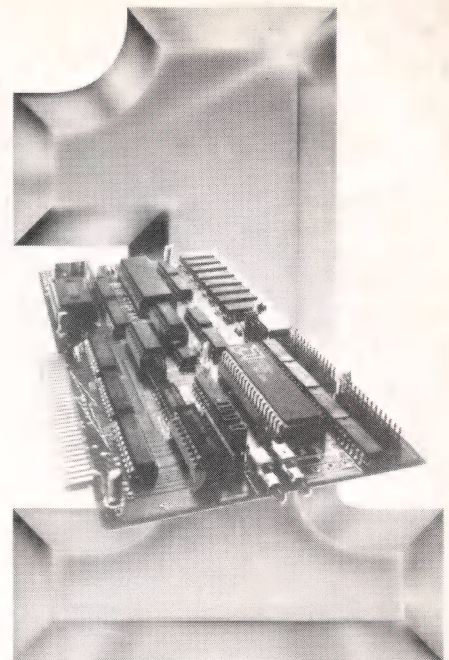


```

303:          wchar( *bp++ );
304:          break;
305:
306:      case END:          /* Go to right extremity */
307:          if( bp < maxbp )
308:          {
309:              ptail( bp, maxbp, 1 );
310:              bp = maxbp ;
311:          }
312:          break;
313:
314:      default:
315:          /*      If we aren't at the right-most extreme
316:          *      of the buffer, move the tail over to
317:          *      make room for the current character,
318:          *      else just print it.
319:          */
320:
321:          if( bp <= end && ( ' ' <= c && c < 0x7f ) )
322:          {
323:              if( bp < maxbp )
324:              {
325:                  if( maxbp < end )
326:                      maxbp++;
327:
328:                  movmem( bp, bp+1, end-bp );
329:                  ptail( bp, maxbp, 0 );
330:              }
331:              wchar( c );
332:              if( bp < end )
333:                  *bp++ = c;
334:              else
335:              {
336:                  /* we're at the */
337:                  wchar( '\b' ); /* right margin */
338:                  wchar( BEL ); /* back up and */
339:                  *bp = c ;      /* ring the bell */
340:              }
341:              if( bp > maxbp )
342:                  maxbp = bp;
343:              break;
344:          }
345:          break;
346:      }
347:
348:      /*      Delete trailing whitespace, terminate the string, go
349:      *      to the next line, and return EOF if we're at end of
350:      *      file, the end pointer otherwise.
351:      */
352:
353:      for( ; *end == ' ' && end >= start ; --end )
354:          ;
355:
356:      *++end = '\0';
357:      wchar( '\r' );
358:      wchar( '\n' );
359:
360:      return ( c == EOF && start == end ) ? NULL : end ;
361: }
362:
363: /*-----*/
364:
365: 371: char *getl( buf, maxline, fp )
366: 372: char *buf;
367: 373: FILE *fp;
368: 374: {
369:     /*      Works exactly like fgets but returns a pointer to the
370:     *      end of the string on success.
371:     */
372:
373:     register int c;
374:     register char *bp = buf;
375:
376:     while( ( c = fgetc(fp) ) != EOF && c != '\n' && --maxline > 0 )
377:         *bp++ = c;
378:
379:     *bp = '\0' ;
380:
381:     return( ( c == EOF && bp == buf ) ? NULL : bp );
382: }
383:
384: /*-----*/
385:
386: 391: char *efgets( buf, maxline, fp )
387: 392: char *buf ;
388: 393: FILE *fp ;
389: 394: {
390:     /*      An editing version of fgets.
391:     *      Works like fgets but recognizes a back-slash at end of line
392:     *      if fp is stdin then raw i/o is used and various editing
393:     *      functions are enabled (see egets for details). A pointer
394:     */

```

(Continued on next page)



**Number One
in Performance**

**Z80H
BLUE STREAK™**

**IBM/AT/XT/PC- 8mz
No Wait States**

FEATURES

- 64K-256K RAM
- 2K-8K EPROM/Static Ram
- 2 Serial Ports
- Async/Sync/Bisync Communications
- Real Time Clock
- Memory-mapped Dual-port BUS
- On-board/Remote Reset NMI capability
- Up To 32 Boards Per AT/XT/PC
- Can Operate As Standalone Processor
- Less Than Full Size Board
(will fit other compatibles.)

SOFTWARE

- ZP/M™ CP/M Emulation Software
(Supports Most CP/M Software)
- Multiuser Capability if Used As A
Slave Processor

IBM is a registered trademark of International Business Machines.
CP/M 80 is a registered trademark of Digital Research Corp.

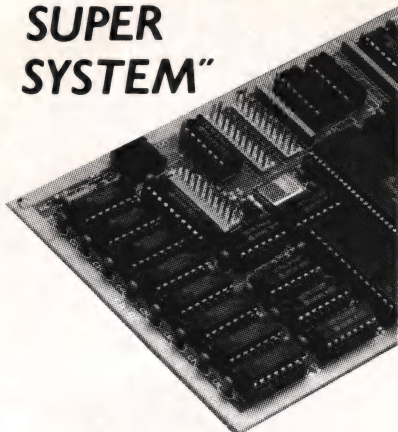


4704 W. JENNIFER, SUITE 105 • FRESNO, CA 93711
209/276-2345

Circle no. 175 on reader service card.

Byte Magazine called it.

"CIARCIA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985,
the SB180 lets CP/M users upgrade to a
fast, 4" x 7 1/2" single board system.

- **6MHz 64180 CPU**
(Z80 instruction superset), 256K RAM,
8K Monitor ROM with device test, disk
format, read/write.
- **Mini/Micro Floppy Controller**
(1-4 drives, Single/Double Density,
1-2 sided, 40/77/80 track 3 1/4", 5 1/4"
and 8" drives).
- **Measures 4" x 7 1/2", with mounting holes**
- **One Centronics Printer Port**
- **Two RS232C Serial Ports**
(75-19,200 baud with console port
auto-baud rate select).
- **Power Supply Requirements**
+5V +/-5% @500 mA
+12V +/- 20% @40mA
- **ZCPR3 (CP/M 2.2/3 compatible)**
- **Multiple disk formats supported**
- **Menu-based system customization**

SB180-1

SB180 computer board w/256K
bytes RAM and ROM monitor
..... \$369.00

SB180-1-20

same as above w/ZCPR3, ZRDOS
and BIOS source..... \$499.00

-Quantity discounts available-

NEW

COMM180-M-S

optional peripheral board adds
1200 bps modem and SCSI/
hard disk interface.

**TO ORDER
CALL TOLL FREE
1-800-635-3355**

**TELEX
643331**

For technical assistance or
to request a data sheet, call:

1-203-871-6170



**Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066**

C Chest (Listing continued, text begins on page 16) Listing Five

```

401:      *   to the end of the input string is returned on success or
402:      *   NULL on end of file.
403:      */
404:
405:      register char  *bp, *start = buf ;
406:      int            linelen, col;
407:
408:      if( fp == stdin )
409:      {
410:          /* linelen is the amount of space left on the
411:           * current input line. Col is the column component
412:           * of the current cursor position.
413:           */
414:
415:          linelen = 80 - (col = getcur() & 0xff);
416:      }
417:      while(1)
418:      {
419:          bp = (fp == stdin) ? egets( buf, min(linelen, maxline) )
420:                           : getl( buf, maxline, fp )
421:                           ;
422:
423:          /* If egets() found and ESC (bp == -1) or we've
424:           * hit end of file (lbp) or we've seen a
425:           * blank line (bp == buf) the last character on the
426:           * line isn't a \, break; Note that in the first
427:           * case we erase then entire buffer.
428:           */
429:
430:          if( bp == -1 )
431:          {
432:              *(bp = start) = '\0';
433:              break;
434:          }
435:
436:          else if( lbp || bp <= buf || *(bp-1) != '\\' )
437:              break;
438:
439:          /* Adjust maxline to compensate for the characters
440:           * already gotten and decrement bp so that we'll
441:           * overwrite the \ on the next pass. Then, if
442:           * we're getting input from stdin, position the
443:           * cursor in its original column but on the current
444:           * line.
445:           */
446:
447:          maxline -= (--bp - buf);
448:          buf      = bp ;
449:
450:          if( fp == stdin )
451:              setcur( getcur() & ~0xff | col );
452:      }
453:
454:      return( bp );
455: }
456:
457: /*-----*/
458:
459: #ifdef DEBUG
460:
461: main()
462: {
463:     static char buf[80] ;
464:
465:     printf("          1          2          3          4\n");
466:     printf("1234567890123456789012345678901234567890\n");
467:     printf(" ");
468:
469:     while( efgets(buf, 40, stdin) > 0 )
470:     {
471:         printf("          %s<---\n", buf );
472:         printf("          1          2          3          4\n");
473:         printf("1234567890123456789012345678901234567890\n");
474:         printf(" ");
475:         *buf = 0;
476:     }
477: }
478:
479: #endif

```

End Listings

Mystic Pascal

Fastest Compiler on Earth—\$64!

Mystic Pascal compiles at 100,000 to over 1,000,000 lines per minute! How? It takes a short cut called *incremental compilation*. Compared to earlier Pascals, the effective speed is astronomical. Give the Compile command for a 1000 line program, and you probably can't lift your finger from the keyboard before the compiler flashes—DONE!

640K of storage not 64K. Are you fed up with being forced to shoehorn your programs into 64K? You won't need mystical powers to run your program in the full 640K that we allow—code, data and stack.

Interactive Pascal. You can enter Pascal statements directly and see the results instantly. It works like a Basic interpreter but it's a true compiler!

Optimized 8086 Code. Mystic Pascal produces true 8086 object code. The TWO code optimizers run in the background so they don't slow you down. Thanks to another breakthrough, our software floating point arithmetic runs faster than Intel says is possible on the 8088/8086 chips.

Real Multi-Tasking Pascal. Advanced programmers may write truly concurrent Pascal programs by simply starting Pascal procedures. Up to 100 concurrent procedures can communicate by passing messages through queues.

ISO Standard Pascal. Educators in particular need a truly Standard Pascal for their students. And learning is made easier by the Help Windows which describe Pascal.

Mystic Canyon Software
P.O. Box 1010
Pecos, New Mexico 87552

Place your order by phone today—
(505) 757-6344 or mail the coupon.
Requires an IBM Personal Computer or true compatible with 256K. Not copy protected.

Rush me the Mystic Pascal System with diskette and 75 page manual!

Name _____

Address _____

City _____ State _____ Zip _____

☐ Check/Money Order

☐ Visa

☐ Mastercard

Price is \$64 total. CODs and Purchase Orders are NOT accepted.

Outside US & Canada add \$15. Payment must be in US funds on a US bank.

Card _____ Exp. _____

Signature _____

Windowing Operating Environments

TopView, GEM,
and Windows

by Michael Swaine

*What's really going on inside GEM,
beneath TopView, behind Windows?*

It's a good thing Bill Gates isn't thin-skinned. Otherwise he might take offense at the way so many programmers are spending so much time enhancing, augmenting, bypassing, subverting, taming, masking, and hiding MS DOS. And now Microsoft is getting into the act with Windows. This article examines Windows and two other "windowing, multitasking programming environments" for MS or PC DOS machines: IBM's TopView and DRI's GEM.

You could argue that the seeds were planted by no less a gardener than the Jolly Blue Giant when IBM released its PC with no operating system but with three operating system options: PC (a.k.a. MS) DOS; CP/M-86; and Sof-Tech's UCSD p-System. Although IBM's subsequent pricing strategy strongly hinted that PC DOS was the PC's DOS, the other operating systems never entirely took the hint and went away. Nor did they even get humble: Consider NCI's recent ads for its p-System that announce a new release for the whole PC family (except for jr, whom the family never mentions). The ads, directed at programmers, suggest that with the p-System you can have compatibility with the operating system that your customers, in their ignorance, want (DOS), but can still work in a good programming environment (not DOS).

Although there were precursors, DRI should probably get the credit for bringing concurrent processing to personal computers with its concurrent version of CP/M. It soon became clear to DRI that to sell concurrency on IBM PCs it had to develop a concurrent PC DOS, acknowledging Microsoft's control of the PC operating system area. But DRI had raised the issue of concurrency on a single-user computer.

After DRI opened the concurrency window, there came the pop-ups, an undisciplined horde of dwarf programs that demonstrated even to ordinary mortals the benefits of some approximation of concurrency. They presented object lessons in the hazards of operating system anarchy as well. It was cute when Eastman Kodak put a package under the tree labeled "Open me first"; it was less appealing to confront a dozen software packages all clamoring "Load me first."

What allowed pop-ups to show off their concurrency was windows, an architectural feature opened by Xerox and polished by Apple. The success of the concept of windows is exemplified by its employment in dWindow, a non-operating system application. For years, Ashton-Tate's dBASE II set the standard for austerity in user interfaces—you can't get much simpler than a single period prompt. Liberty-Bell Software's dWindow does a dazzling cathedral window treatment on dBASE II and III that makes them look like entirely different products. But it is at the operating system level that the fenestration of the interface has become obligatory, and it was Apple's Macintosh that made it so.

Bill Gates and his Microsoft programmers were among the first developers to peek behind the blinds at Apple, and they were not slow to announce Microsoft's own window-oriented product, which they called Windows. Sometime between the announcement of Windows and its release, the Korean War ended, Alaska and Hawaii attained state-

hood, and Digital Research programmers conceived and brought out a windowing product called GEM, whose visual display, Apple recently noticed, looks rather like the Mac's. Meanwhile, IBM caught the drift, fumbled through its pockets, and pulled out a windowing, multitasking operating environment that it christened TopView.

With at least three windowing, multitasking operating environments to load on top of, stack next to, or wrap around DOS, the archetypal "fully loaded PC" was beginning to look like a pop-up on steroids. In fact, or at least in *PC Week*, Peter Norton described a dream he had in which he tried to run TopView, GEM, and Windows simultaneously. His hard disk flew to Poland.

Smart Money Talks a Lot

A programmer must face more serious questions of compatibility. How does each product work with DOS? How will existing and future programs run in each of these environments? How well will they run? What are the costs of laying another sheet of software between programmer and machine? We asked such questions of development-team members for each of the three products. But as the answers came back, industry analyses were circulating that questioned the viability of at least two of the products (the two that had actually been released). Were we wasting our time asking questions about these products?

Although some developers are producing TopView-compatible applications, the product has clearly not captured the imagination of users or developers. The recent agreement between IBM and Microsoft has fueled much speculation about the viability of TopView, and many observers seem to be betting on Microsoft. One popular argument is that IBM will create two systems: one using Windows and Microsoft's DOS, the other using a DOS-free descendant or unrelated successor to TopView—in any case, a proprietary operating system. Some, including IBM watcher Andy Seybold, speculate that the IBM-Microsoft agreement may have little to do with IBM's long-term goals and that IBM will use TopView as the centerpiece of its future operating systems. At least one analyst insists that IBM's history demonstrates that it will eventually have a proprietary operating system on the PC. But IBM has broken tradition several times in the short history of the PC, and the venerable wisdom about what IBM always does may not be as wise now as once it was. It does seem that it is getting harder to second-guess IBM, and that the benefits from succeeding therein may be even more dubious today than in mainframe days.

The smart money is writing off GEM in the IBM-compatible market, reasoning that if IBM owns the hardware and Microsoft owns DOS, where does that leave a company that tries to compete with them on their turf?

Perhaps DRI's success with GEM will have to come in the other end of the dumbbell. Lee Felsenstein, in proselytizing for his Hacker's Mac project, describes the personal computer market as an asymmetric dumbbell with one globe of IBM-compatibles and a smaller globe of Macalikes. He argues that the smaller globe can grow and prosper only with compatibility and proposes a radical strate-

gy for forcing compatibility on unwilling Apple, Commodore, and Atari. Such a development would likely benefit DRI and GEM, but is it likely?

Perhaps not, given the fact that Apple has pressured DRI into changing GEM to decrease its similarity to the Macintosh's visual interface. GEM as originally released had to be terminated by November 15, that is, DRI had to stop supporting and advertising it. The new version will look less Macintoshish. Apple is also talking with Microsoft about Windows, but in softer tones.

Smart money and dream-machine designers aside, GEM provides MS DOS users and programmers with capabilities that Windows and TopView lack. But TopView and Windows have their own distinct, desirable features.

Three facts argue against accepting the judgment of the smart money too hastily. First, the three operating environments offer three different sets of capabilities to users and software developers. Second, no one needs any of these products; they are all frosting on the DOS, and users may decide among them—or against all of them—on grounds that smart money would consider dumb. Third, as Bob Frankston pointed out in *InfoWorld*, writing for TopView (or Windows or GEM) limits your market. In any case, we assume there is merit in understanding these products from a programmer's point of view.

Inside GEM

At one level, getting started developing an application that is compatible with GEM is simple. You get the Programmer's Toolkit and start writing. In terms of hardware, you need a PC with half a megabyte of memory and a color-graphics adapter. You should also have a hard disk and a mouse, although you can do development work without them.

Beyond these elementary requirements, one comes up against the fact that GEM is a message-passing program. Years of single-thread procedural programming experience will not prepare you adequately for the different programming model that GEM employs. According to one GEM programmer, the learning curve within DRI during GEM development was two months. Programmers who had never worked with any windowing system took two months to get up to speed in the GEM programming environment. Programmers who had had some windowing experience learned faster but had to unlearn some details that did not transfer.

GEM itself does not communicate directly with DOS; that is, the Virtual Device Interface and Window Services don't. The GEM Desktop does. File manipulation is handled through DOS calls. GEM supports a variety of devices and is expected to support the AST/Ashton-Tate/DRI/Quadram expanded-memory specification, which will become particularly significant when the multitasking version of GEM is released.

GEM compares more directly with Windows than with TopView; TopView is character-oriented and truly multitasking and GEM and Windows are neither. In comparing GEM and Windows as programming environments, Windows developers point to powerful features, and GEM de-

velopers talk about a clean programmer's interface. But the clearest advantages GEM has over Windows at the moment seem to be that GEM has been out long enough to land significant committed and producing OEMs, including Atari and Apricot, and the leverage that this gives the developer in porting an application to different environments.

What does the Apple-DRI settlement mean to programmers who have developed GEM-compatible software or who are thinking about doing so? Perhaps not much. The changes in GEM forced by Apple appear to be essentially cosmetic, and though the impact on the GEM Desktop will be significant, the programming impact may be minimal.

Beneath TopView

The chief difference between TopView on the one hand and GEM and Windows on the other is that TopView is truly multitasking. You can see multitasking in action if you bring up a visually active BASICA program in two TopView windows simultaneously. You'll see the program doing its thing in parallel with itself.

Because it is a multitasking operating environment, TopView permits the development of multitasking applications. The application developer can produce a task with its own subtasks. Then, in addition to running the application in multitasking mode with other tasks, TopView will multitask the application's subtasks. Tasks can communicate; if you get the object handle for a task you can send it a message.

Memory management under TopView is as simple as it is under DOS. When a program begins, it is assigned memory according to the demands of its fixed Program Information File. Henceforth it can get no more nor less. On a program's termination, its memory is freed.

TopView does not replace DOS. It sits atop DOS, handing off file I/O and other system functions to the system. It is at the character I/O level that TopView butts in, redirecting all character I/O. Among other things, this means that you can freely mix TopView and DOS calls. You can put the user in DOS under TopView and have the fact that TopView is active be unknown to them. (A Getversion call will show the user that TopView is active.)

Although TopView is character-oriented rather than graphics-oriented, it will work with a standard monochrome or color screen and anything up to the EGA. TopView will work with the EGA but won't take advantage of its extended graphics capabilities.

TopView developers are trying to woo other developers, arguing that you can bring a more powerful application to market more quickly if you write for the TopView environment—you can take advantage of a standardized display style, a toolkit of window design aids, and other development tools. You can also use what TopView designers call full-screen input. This wooing of developers has not been entirely unsuccessful; the Trio micro-to-mainframe product and the Lattice Topview Toolbasket are significant TopView-compatible products.

For existing programs, you can make use of as much of the windowing capability as makes sense. For some appli-

cations this may be useless, but the capability is there. The implementation of DOS services shows what IBM projects for future applications—making logical use of several windows rather than just throwing a frame around a full-screen display.

Behind Windows

Because Microsoft has good sources on what future versions of DOS will require, its programmers can do what they admonish independent developers not to do—work around DOS. Windows runs "side by side with DOS." When Windows comes up, its DOS executive replaces command.com, which is no longer needed.

Windows is made up of three pieces: two are the external pieces that everyone sees, the user interface and the graphics device interface (GDI). The third is the kernel, which interfaces to whatever kind of MS DOS is on the machine. This interface changes considerably with different versions of DOS and hides the operating environment from Windows.

The user interface differs from the Mac/GEM/TopView approach in tiling the screen with windows that don't overlap other windows.

At the GDI level is an interrogating interface: if a device says that it can only do bitblt, the interface will simulate everything else in the software using bitblt; if the device says it can do complex polygons with hash filling, GDI hands off to the device. The device manufacturer fills in capabilities up to some level and Windows simulates the rest. The goal at the GDI level was device independence—the ability to change one line of code and have output go to a different device.

At the level of the kernel, Windows has true compacting global memory management; it does not have true preemptive multitasking. It can allocate, reallocate, dynamically free and restore data. Its task handling is round robin nonpreemptive multitasking—the application must yield control. There are mechanisms for implicit yielding when the application is waiting for something.

The reason Microsoft gives for not implementing true preemptive multitasking is suggestive: DOS is not reentrant. If you preempt a task and it's in DOS at the time, you'd best get back to DOS quickly. So a good chunk of the time slicing would not be beneficial. That, Windows developers claim, is why TopView is slow. Rick Dill of the Windows design team says, "We think multitasking really belongs at the operating system level, and it will get there eventually." He means in MS DOS.

Windows started from the Smalltalk push model and changed to a more procedural approach only when Microsoft found that programmers didn't work well with its implementation of the push model. The general structure of a Windows application is: initialization; Windows-required initializations (because Windows does not require that an application be installed it does require that the application register itself with Windows when it starts); creating a window; and the main program while loop—Getmessage, Translatemessage, Dispatchmessage.

One of the hidden features is the ability for applications

Combine The Power Of C And Your IBM Mainframe

What are the advantages of running C on your 370?

C is a high level language offering speed, power and flexibility. When you combine this with the IBM 370, you've got an unbeatable environment for software development. In addition, C lends itself to many different types of applications, is highly portable to a number of diverse CPU architectures and is the basis for a wide variety of existing software. Clearly, the future of programming lies with C.

Why Choose Whitesmiths' C?

Experience. Whitesmiths has been developing C compilers for over seven years. We were one of the first to introduce C for the 370 and our experience is evident in the features we offer:

- Full implementation of the C programming language for the IBM 370
- Ability to run under VM/CMS and MVS, and to run interactively under TSO
- Ability to produce re-entrant code (discontiguous shared segments)
- Direct support for both EBCDIC and ASCII character sets
- Optional cross support for MS/PC-DOS, CP/M-86, CP/M-68K, VERSAdos-68K and CP/M-80
- Unlimited use of libraries in binary form
- Support for ROM-based programs
- UNIX-style compatible library
- Optional use of IBM or Whitesmiths assembler
- Support of full ISO Level 0 Pascal, extended to support separate compilation
- System and Machine Interface Libraries provided in source code form



Call Whitesmiths' toll-free number for more information
1-800-225-1030

INTERNATIONAL DISTRIBUTORS: UNITED KINGDOM - Real Time Systems, Ltd., Douglas, Isle of Man, 011-44-0624-26021; AUSTRALIA - Neology Pty. Ltd., Rosebury, Sydney 662-4111; JAPAN - Advanced Data Controls Corp, Toshima-Ku, Tokyo, 03-576-5351; SWEDEN - Unisoft AB, Goteborg, 011-46-31-125810; FRANCE - Cosmic SARL, Paris, 011-33-378-8357

Whitesmiths, Ltd. • 97 Lowell Road • Concord, MA 01742 • 617-369-8499 • Telex 750246

New C Tools

***SPECIAL OFFER \$349**
UniTools, "Z," PC-LINT and
Sunscreen.

UniTools \$149 \$99
MAKE, DIFF and GREP

These versatile UNIX utilities put power at your fingertips. MAKE, a program administrative tool, is like having an assistant programmer at your side. DIFF compares files and shows you the differences between them. GREP can search one or many files looking for one pattern or a host of patterns.

"Z" \$149 \$99

A Powerful "vi"-type Editor
 Based on the Berkeley "vi" editor, "Z's" commands are flexible, terse, and powerful; macro functions give you unlimited range. Features include "undo," sophisticated search and replace functions, automatic indentation, C-tags, and much, much more.

SunScreen \$149 \$99

Low-priced screen utility
 Create and modify formatted screens easily. Validate fields, support function keys, color and monochrome cards. With library source SunScreen is \$199.

PC-LINT \$149 \$99

Error checking utility
 A LINT-like utility that analyzes programs and uncovers bugs, quirks and inconsistencies. Detects subtle errors. Supports large and small memory models, has clear error messages and executes quickly. Has lots of options and features that you wouldn't expect at this low price.

The support products listed here run with most leading MS/DOS and PC/DOS C compiler systems, including **Manx Aztec**, Lattice, Microsoft, Mark Williams and Computer Innovations C.

Learn C Today!

C' PRIME \$99

***INTRODUCTORY OFFER! \$99**
C' Prime, Personal Computing
and C, Plus APPRENTICE C

Never has C been easier to learn. **Manx Aztec** is now offering a complete C system called **C' Prime** at an exceptionally low price. This powerful system includes a Compiler, Linker, Assembler, Editor, Libraries and Object Librarian. **C' Prime** has full UNIX compatibility, and supports a host of third-party software.

NEW FROM ASHTON-TATE!

A Practical Step-by-Step Introduction to C Programming
 written Especially for use with **Manx Aztec's new C' Prime:**

Personal Computing and C, is a detailed, easy-to-understand guide to C prepared especially by Ashton-Tate for use with the new **Aztec C' Prime**. Written by John Gainsborough of NASA's Jet Propulsion Laboratory, it includes chapters on C programming basics, function libraries, data handling, and many other functions. The book also includes a complete demonstration program of an accounting database update application, as well as dozens of examples on how you can program effectively with C.

APPRENTICE C

Learn C with this easy to use, and complete C language interpreter. APPRENTICE C includes a complete one-step compiler that executes with lightning speed, an editor, and a run-time system. **\$49.95**

Tools for C

C Utility Library \$185 \$135

Over 300 C subroutines
 C and assembler source code for screen handling, color printing, graphics, DOS disk and file functions, memory management and peripherals control. Graphics and speed are impressive.

C-tree \$395 \$309

B-Tree database system
 Store, update and retrieve records easily. Both high-level multi-key ISAM routines and low-level B-Tree functions. Available for MS-DOS, CP/M-86, CP/M-80 and others. Easily transported. Adaptable for network and multiuser. Includes source.

PHACT \$250 \$200

Data Base Record Manager
 Includes high-level features usually found in larger database systems. Tunable for speed or size optimization. Available for MS-DOS, CP/M-86 and CP/M-80.

Pre-C \$295 \$315

LINT-like source code analyzer
 Locates structural and usage errors. Cross-checks multiple files for bad parameter declarations and other interface errors.

Windows for C \$195 \$159

Versatile window utility
 Supports IBM PC and some non-compatible environments.

PANEL \$295 \$235

Screen generating utility
 Create screens via simple, powerful editing commands. Select colors, sizes and types, edit fields. Directly input data to a multi-keyed file utility, and tailor screens for each application.

HALO \$250 \$199

Ultimate C graphics
 A comprehensive package of graphics subroutines for C. Supports multiple graphics cards.

PLINK-86 \$395 \$319

Overlay linker
 Includes linkage editor, overlay management, a library manager and memory mapping. Works with Microsoft and Intel object.

To order or for information call:

TECWARE

1-800-TEC-WARE

(In NJ call 201-530-6307)

written by different vendors to be integrated via cut and paste as well as the via the messaging mechanism. For example, Multiplan could register a message that says "cell changed." A presentation-graphics program could know that that message means it can look in the clipboard for the contents of the cell and automatically update the chart being displayed.

Windows is 85 percent written in C; the low-level part is written in assembly. Windows supports Pascal, FORTRAN, C, or Microsoft's macro assembler. Because of the new .exe format, programmers will have to use Microsoft's new linker.

According to Dill, Microsoft hopes to entice software developers with the knowledge that, in conforming with Windows, they're building a future DOS-conforming application. Windows is a sneak peek into DOS. (On the other hand, Dill admits there are software developers who have disassembled the .exe file and know more about the innards of DOS than he does.)

But does all this just get in the software developer's way? Most serious software developers find that they must work around DOS to produce a competitive product. Won't software developers find Windows just as encumbering?

"Our bitblt," Dill responds, "is faster than yours." He's serious: Microsoft's bitblt is a general-purpose source-pattern-destination function supporting four operators. The result is that there are 256 different operations avail-

able: others implement some subset, often 16, of those. The Windows graphics routines can move a graphics block around the screen almost as fast as the data can be moved through memory. Whether that answers the question is moot.

One significant change Microsoft is introducing is a new .exe file format. Future versions of DOS (of which Windows is a hint) will need more information in the .exe file. To maintain some compatibility, Microsoft has grafted the old .exe header, code, and data on top of the new .exe header. The old tells where the new begins, and the new has its own code and data. It also has something totally new called resources. Resources (like menus and dialogue boxes) can be changed without messing with the .exe file. Thus, one can have one binary that works worldwide, with the resources supplying the language-specific information.

What, then, of the .com files? In the next version of DOS, .com files go away. DOS 3 is the last Microsoft operating system in which memory-image programs will be supported under the operating system itself. "We just need the information," Dill explains, "to be able to do things like running that application up in high memory and being able to segment it correctly and deal with running an old style 8088 unprotected application under a 286."

DDJ

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 192.

MORE... VALUE and PERFORMANCE

with Mitek's Relocatable Z80 Macro Assembler and Z80 Symbolic Debugger

NEW Z80 Symbolic Debugger

- Only \$49.95 plus shipping.
- Screen oriented with a simultaneous display of instruction mnemonics, register, stack, and memory values.
- Breakpoints may be set on any combination of fixed memory address, register values and/or memory values.
- Uses Digital Research compatible SYM files.
- Supports Hitachi HD64180.

Relocatable Z80 Macro Assembler

- Only \$49.95 plus shipping.
- 8080 to Z80 Source Code Converter.
- Generates Microsoft compatible REL files or INTEL compatible hex files.
- Compatible with Digital Research macro assemblers MAC & RMAC.
- Generates Digital Research compatible SYM files.
- Conditional assembly.
- Phase/dephase.
- Cross-reference generation.
- Full Zilog mnemonics.
- INCLUDE and MACLIB FILES.
- Separate data, program, common, and absolute program spaces.
- Supports Hitachi HD64180.
- Z80 Linker and Library Manager for Microsoft compatible REL files available as an add-on to Assembler.

ATTENTION Turbo Pascal Users:

Assembler will generate Turbo Pascal in-line machine code include files.

PRICE LIST

Z80 Macro Assembler: \$49.95
Assembler, Linker, and Library Manager: \$95.00
Manual Only: \$15.00
Z80 Symbolic Debugger: \$49.95
Manual Only: \$15.00
Assembler, Linker, Library Manager, and Debugger: \$134.95
Include \$5 for shipping and handling.

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: (808) 623-6361

Specify desired 5 1/4" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.

P. O. Box 2151
Honolulu, HI 96805

MITEK

Z80 is a trademark of Zilog, Inc. MAC and RMAC are trademarks of Digital Research, Inc. Turbo Pascal is a trademark of Borland International, Inc.

Circle no. 71 on reader service card.

BANKSWAP

A Banked Memory Debugging Tool for CP/M Plus

by Albert S. Woodhull

By making extended memory useful to 8-bit processors, CP/M Plus can help to keep 8080-, 8085-, and Z80-based microcomputers in competition with newer 16-bit systems. Although the 8-bit processors can directly address only 64K of memory, under CP/M Plus a megabyte can be efficiently used.

For the programmer there is a catch. Transient programs must reside in a single 64K bank. They have indirect access to alternate banks through the operating system. A user running purchased software will not be bothered by this. Developers and others who need to write or modify code that uses the alternate banks have a serious problem, however. CP/M Plus comes with development tools, such as the RMAC assembler and

ability for a user program even though the program resides entirely in the main bank. The speed increase comes from using the extra memory to buffer disk data and disk directories. The space increase comes from putting part of the operating system in an alternate memory bank, making almost all of the 64K of memory the processor can address available to the user's program. For instance, on an Apple with the Advanced Logic Systems CP/M Card the transient program area (TPA) is 60K, even though the CP/M Plus system itself (exclusive of the CCP) takes up 17K.

User programs cannot directly access data or executable code in alternate banks. The point can be argued, but I believe Digital Research made a wise choice in imposing these restric-

Apple's Filer is fine until you want to copy files between volumes without losing a BASIC program in memory.

LINK, that enable one to write code for use in any bank of memory, but the standard CP/M Plus debugging tool, SID, is an ordinary transient program, incapable of examining or changing the alternate banks. The programmer needs new tools or a way to extend the power of the old ones.

My response to this need was BANKSWAP, an extension for SID or DDT that provides access to the alternate banks. I will describe this tool after a brief discussion of how CP/M Plus uses banked memory.

Banked Memory in CP/M Plus

Additional banks of memory can improve the speed and memory avail-

ties. Most programs that use a lot of memory use it to hold data, and disk buffering can provide an improvement in data access comparable to that of extended memory. Given the inherent limitations of 8-bit processors, it is not clear that allowing user programs to execute code resident in extended memory would be much more efficient than using code overlays swapped in from disk buffers.

Under CP/M Plus, programs must be written as if the extra memory is not there at all. As long as the minimum amount of memory needed for an application is present in the TPA bank, a program will be able to run on any CP/M system. This continues the CP/M tradition of providing portability for programs by using the operating system to make details of hardware irrelevant to user programs.

Albert S. Woodhull, Hampshire College, Amherst, MA 01002

The BANKSWAP Program

BANKSWAP is not a stand-alone program; it is an enhancement to SID or DDT that provides additional commands to copy blocks of memory from bank to bank. The normal functions of the debugger can be used on a copy of memory from another bank that has been brought to the TPA bank.

BANKSWAP is relocatable and is not necessarily loaded to the same location each time it is used. For ease of use, BANKSWAP installs a vector to its own entry point at the RST 5 location (28H) during installation. Typing G28 from the SID or DDT prompt brings up the BANKSWAP menu. The initialization process also displays a message to remind the user of the presence of BANKSWAP and the command to access it.

The BANKSWAP menu allows the user to choose the direction of the move, the memory addresses for the source and destination, and the length of the block to be moved. The menu also provides for easy return to DDT or SID and for the eventual removal of BANKSWAP. Copying is done in two steps through a buffer that is also in common memory. I chose to use a relatively small buffer and repeat the process several times in order to make the best use of memory space.

Listing One (page 38) for BANKSWAPASM contains comments that explain the operation of BANKSWAP, but I will emphasize a few points I found important in working with banked memory. Although I wrote this program for use on an Apple with the Advanced Logic Systems CP/M Card, there should be no problems in making BANKSWAP work on other implementations of CP/M Plus. The most critical point is to be sure that control is not lost while bank 1 is deselected. This means ensuring that BANKSWAP itself, the stack, and all data areas used are located in common memory. It is possible for a CP/M Plus system to be constructed so that interrupts and system calls can be handled while alternate banks are selected. With insurance in mind, I thought it best to disable interrupts and avoid calls to the standard BDOS entry point while bank 1 is deselected because the vectors on page 0 of bank

1 are then inaccessible.

I used a direct call to a BIOS routine to select the bank of memory. There are several points to mention in regard to that. First, the CP/M Plus documentation is emphatic in stating that direct BIOS calls must not be made by application programs; the reason for this is that under CP/M Plus some BIOS routines are always called from the bank 0 portion of the BDOS and will not return to a program located in bank 1. In fact, a separate BDOS function is provided for gaining direct access to the BIOS. The

catch is that this BDOS function prohibits access to one BIOS routine—you guessed it—the one we need for BANKSWAP. Digital Research really doesn't want user programs to try to access other memory banks.

Having decided to take the law into your own hands and call the BIOS bank-select routine directly, you can't do it by accessing the BIOS vector in low memory just any old time for the same reason you can't use BDOS routines any old time—some of the time the low memory the program sees will be in bank 0. For

If you use an IBM PC, you're stuck with DOS.

But that doesn't mean you have to develop under DOS. Everybody knows that IBM and DOS go together like the old-fashioned horse and carriage. Like it or not, DOS dominates the PC market and until now, developers have just had to grin and bear it.

But now you have a choice. NCI's UCSD p-System is a powerful solution for the developer who requires an outstanding development environment but wants to retain compatibility with DOS. The UCSD p-System supports the full range of ANSI Standard Pascal instructions, plus several valuable extensions. It supports large libraries of separately compiled modules and code overlays, so that you can develop larger programs than ever before, as well as a wealth of development tools including a full-screen editor, a file manager, a library manager, full interrupt serial support, TurtleGraphics, automatic startup file, and many other software utilities. At the same time, you can send out your application on a DOS diskette, boot from DOS, use DOS hard disk partitions and automatically read from and write to DOS files.

So if you want a development environment that gives you speed, flexibility, efficiency, provides truly incredible programming capability, yet still gives your end-user the familiarity of DOS, choose NCI's UCSD p-System -- the powerful solution.

Order your p-System solution today: call (604) 430-3466.



Network Consulting Inc.

#115-3700 Gilmore Way, Burnaby, BC, Canada V5G 4M1
US Mailing Address: P.O. Box 8040, Blaine, WA 98230

Circle no. 168 on reader service card.

this reason, I made the initialization portion of BANKSWAP fetch the BIOS vector so the necessary entry point could be calculated and stored locally. I did this only for the BIOS SELMEM routine; the program can be simplified if a BIOS XMOVE routine has also been implemented (this is not provided in the Advanced Logic Systems release of CP/M Plus).

Resident System Extensions

BANKSWAP must run in common memory. This is most easily accomplished by creating a relocatable file

to be loaded to the top of the TPA. CP/M Plus has another feature, the Resident System Extension (RSX), to simplify this task.

An RSX is a page-relocatable program segment that CP/M Plus loads to the highest available address, just as DDT or SID are handled under older versions of CP/M. Each RSX module has a prefix, as shown in Listing One. The prefix includes a jump to the BDOS entry point in high memory (for the first RSX installed) or to the last previously installed RSX. Whenever an RSX is installed, the JMP in-

struction at location 5 is modified to point to it, so a chain of JMPs is traversed whenever a BDOS call is made. The address portion of the JMP at location 5 is also used by programs to detect the decreased size of the TPA, protecting the RSX from being overwritten.

Because all BDOS calls pass through the RSX JMP chain, RSXs can be written to intercept BDOS calls to customize BDOS performance. BDOS interception is not necessary, however. BANKSWAP intercepts only the very first BDOS call after it is loaded—this is a convenient way to force execution of the BANKSWAP initialization phase. The initialization code saves and restores whatever information is being passed to the BDOS and modifies the JMP chain to prevent reinitialization.

An RSX must be connected to a normal .COM program to be installed. CP/M Plus provides a utility, called GENCOM, to do this and also provides the RMAC and LINK programs needed to produce the relocatable program and its relocation bit map. The process is considerably more complicated than assembling and loading a program under CP/M 2, but it is easy to use the CP/M SUBMIT program to direct the process. An RSX can be attached to any .COM file. I have attached BANKSWAP to DDT.COM and SID.COM and it works with both. Listing Two (page 52) shows the RSXMAKER.SUB file that can create BANKSWAP.RSX and connect it to SID.COM.

When BANKSWAP is to be attached to SID or DDT it is assembled with a REMOVE flag set in the RSX prefix. This ensures that when a warm boot occurs upon exiting from SID the space occupied by the RSX will be freed. BANKSWAP can also be assembled to be loaded independently of SID by setting the ALONE equate true. In this case it is still necessary to attach the RSX code to a .COM file, which could be nothing more than a jump to location zero. As the Listing One shows, setting ALONE true resets the REMOVE flag in the RSX header, and it also adds an option to the BANKSWAP menu to allow later removal.

New Release

C

+ UTILITY LIBRARY = PRODUCT

• We have over 300 complete, tested, and, documented functions. All source code and demo programs are included.

• The library was specifically designed for software development on the IBM PC, XT, AT and compatibles. There are no royalties.

• Over 95% of the source code is written in C. Experienced programmers can easily "customize" functions. Novices can learn from the thorough comments.

We already have the functions you are about to write

Concentrate on software development—not writing functions.

THE C UTILITY LIBRARY includes:


- Best Screen Handling Available • Windows • Full Set of Color Graphics Functions • Better String Handling Than Basic • DOS Directory and File Management • Execute Programs, DOS Commands and Batch Files • Complete Keyboard Control • Extensive Time Date Processing • Polled ASYNC Communications • General DOS BIOS gate • Data Entry • And More •

• The Library is compatible with: Lattice, Microsoft, Computer Innovations, Mark Williams and DeSmet. Available Soon: Digital Research, Aztec and Wizard.

C Compilers: Lattice C—\$349, Computer Innovations C86—\$329, Mark Williams C—\$449.

C UTILITY LIBRARY \$185. Special prices on library & compiler packages.

Order direct or through your dealer. Specify compiler when ordering. Add \$4.00 shipping for UPS ground, \$7.00 for UPS 2-day service. NJ residents add 6% sales tax. Master Card, Visa, check or P.O.



ESSENTIAL SOFTWARE, INC
P.O. Box 1003 Maplewood, New Jersey 07040 914 762-6605

Circle no. 138 on reader service card.

There is a potential problem with the version of BANKSWAP shown in Listing One, but I'm leaving it for someone else to fix. As noted above, BANKSWAP uses the RSX technique in order to be located at the highest available memory address. The actual amount of high memory that is common depends upon the hardware used, and I didn't figure out a way for a program to determine this. If too many RSXs are installed, the top of available memory can be below the common region. In this case BANKSWAP will probably cause a crash. On my Rev. A CP/M card the common-memory limit is at 8000H, which leaves room for a lot of RSXs, so I have never given the problem a high priority.

Using BANKSWAP

Once BANKSWAP has been attached to a copy of SID.COM it will be installed whenever SID is run. Listing Three (page 52) shows part of a typical session with enhanced SID. The initialization process tells the operator that BANKSWAP is available. A G28 command enters BANKSWAP, which displays a menu. I usually bring part of an alternate bank over to bank 1 first, then return to the debugger to examine, disassemble, or alter the copy. In fact, if executable code from some portion of bank 0 is brought over to the corresponding address range in bank 1 the debugger can be used to trace through it—providing that the code doesn't switch banks or access I/O or storage addresses in an alternate bank.

In the Apple environment there is one limitation: an Apple's I/O is all memory mapped in bank 0. If an attempt is made to access bank 0 in the range 6000H to 67FFH it may crash the system because some addresses in this range activate switches on the Apple main board or peripheral cards, including the CP/M Card itself.

Conclusion

One of my first reactions to CP/M Plus was a helpless feeling. I had written my own BIOS for CP/M 2.2 on my S-100 system, and I was accustomed to being able to explore how the system worked on any CP/M machine. With CP/M Plus, portions of

the operating system were totally inaccessible to the tools I had on hand.

Most users don't need the kind of access I wanted. However, the initial version of CP/M Plus I received did not support my printer interface properly. I also had proposed to develop CP/M Plus driver software for companies that manufactured large-format disks and RAM-disk add-ons for Apples. The usability of my system as well as potential income depended upon my ability to patch various devices into CP/M Plus.

BANKSWAP solved my problems. It gave me a way to satisfy my curios-

ity about how the Advanced Logic Systems CP/M Plus BIOS worked, and I was able to figure out how to patch some of my peripheral drivers into CP/M Plus. Finally, I now had the tool I needed to begin to develop and debug enhancements to the supplied BIOS. In the process of developing BANKSWAP I also learned how to create and use RSXs, which are very useful CP/M Plus features.

DDJ

(Listing begins on next page)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 193.

DEBUG FAST!

As a professional, your time is valuable. Advanced Trace86™ is designed as a debugging environment to increase productivity. In many situations, Advanced Trace 86™ can pay for itself in one or two debugging sessions.

Advanced Trace86™

FEATURES

- Trace Screen with user-configurable windows for disassembled code, registers, flags, the stack, the 8087 registers, and memory.
- Single-step tracing with virtually unlimited **back tracing**.
- In-line assembly capabilities, edit .EXE and .COM files in memory (including non-destructive line insertion for .COM files).
- Symbolic debugging capabilities - read or create line labels and comments, read variables.

Debugger Comparison Chart - Selected Features

Features	AT86 Code. Per.		
1. Load Program as memory resident (like Sidekick)	X		X
2. IBM Professional DEBUG & Periscope boards supported	X		X
3. Keyboard break-out support - Ctrl-Enter key combo	X	X	
4. Conditional breakpoints	X		X
5. Hex/Decimal calculator & converter	X		X
6. Convert numbers to binary	X		
7. ASCII chart (pop up)	X		
8. In-line Assembler with code insertion capability	X	X	
9. 80286/80287 support in Assembler/Disassembler	X		
10. Protect Virtual Mode Support for 80286	X		
11. 8087/80287 window - registers in both decimal & hex	X		
12. Display memory in ASCII, byte, word, & double word	X		X
13. Define "memory structures" for display/editing	X		
14. Try out DOS interrupts command	X		
15. Pop-up Help windows	X		
16. Command macros defined & saved to disk	X		
17. DOS TYPE, DIR, & ERASE commands	X		
18. Search memory for Assembler code - multiple instructions	X		
19. Execute/trace program in reverse & restore machine state	X		
AT86 - Advanced Trace86	Code. - Codesmith	Per. - Periscope	

Requires: IBM PC or compatible, minimum 128K RAM. Retail Price - \$175.00

Morgan Computing Co., Inc.

P.O. Box 112730 Carrollton, Texas 75011 (214) 245-4763

Listing One

```

; BANKSWAP.ASM
;
; A. S. Woodhull          28 June 83
; rev   1 July 85        -- minor editing
;      20 Oct 83
;
; This is designed to be run as a subprogram under DDT
; or SID, in a banked version of CP/M 3.0 The function
; of BANKSWAP is to copy blocks of memory from other
; banks to and from bank 1, using a buffer in the common
; area. Normal DDT or SID functions can then be
; performed, using the copy of the code in bank 1.
; (Of course, you cannot trace through program segments
; that switch banks or access memory mapped I/O in
; another bank.)
;
; BANKSWAP must reside in the common area of memory, and
; a buffer area through which data can be copied must
; also be present in the common area. BANKSWAP is to be
; assembled as a Resident System Extension (RSX), which
; will automatically be relocated to the top of the TPA.
; It is assumed that the common area is large enough to
; allow an RSX to fit--if this is not true BANKSWAP will
; not work in its present form.
;
; Because the location of BANKSWAP in memory is not fixed
; a jump through a fixed location is set up when the BANK-
; SWAP code is installed. In this version the RST 5 vector
; at 28H is used, but any convenient location on page zero
; may be used.
;
; Under CP/M 3 direct access of BIOS routines is generally
; to be avoided by user programs, since BIOS routines may
; have expectations about which bank is selected when they
; are called. We will, however, do bank switching through
; the BIOS selmem routine. For generality we will use the
; BIOS vector at location 1.
;
FFFF = true: equ 0ffffh
0000 = false: equ not true
;
0000 = alone: equ false ;make false if attached to DDT/SID
;
0001 = biosv: equ 1 ;address of wboot in BIOS found here
004E = selmem: equ 4eh ;offset from wboot
;
0100 = buflen: equ 100h ;move block size
;
; default parameters
1000 = movcnt: equ 1000h ;to move 16 pages (4K) at a time
0100 = b0dft: equ 100h ;start of block in bank 0
0100 = b1dft: equ 100h ;start of block in bank 1
;
; zero page addresses
0005 = bdos: equ 5 ;bdos entry point
0028 = rstv: equ 28h ;RST 5 used as entry vector
;
; BDOS functions used
0001 = conin: equ 1 ;get a char
0009 = printf: equ 9 ;print a string
000A = rdbuf: equ 10 ;read a line
;
; This is standard prefix for a Resident System Extension
; see CP/M 3 Programmer's Guide, 1st ed., sec. 4.4, p.168
;
0000 000000000000 serial: db 0,0,0,0,0,0
0006 C3A203 start: jmp install ;one-time routine
0009 C30000 next: jmp 0 ;altered at installation

```

(Continued on page 40)

NEW LANGUAGE BREAKS OLD RULES. GIVES PROGRAMMERS POWER, SPEED AND SIMPLICITY.

Try this remarkable language, PROMAL™, for 30 Days AT NO RISK and...

We think you'll be thrilled with this breakthrough system when you discover its power, ease of use, and dazzling performance on your IBM PC, Apple IIe/IIc, or Commodore 64. **But we don't expect you to accept our claims for PROMAL without proof**, so we invite you to explore the power of PROMAL on your own during our 30-day trial period.

Broken Rules

Now that PROMAL 2.0 has broken the rules, a structured language doesn't have to be slow, unwieldy and difficult to use. PROMAL is fast, elegant, and simple.

What Is PROMAL?

PROMAL stands for **PRO**grammer's **Micro Application Language**. But PROMAL is more than a high-level language, it's a total structured programming development system with a fast, one-pass compiler, a versatile full-screen editor, plus an integrated machine-language subroutine library. And for APPLE and Commodore systems it includes a DOS-like system "Executive."

Better By Design

PROMAL was designed from "scratch" for optimum performance and ease of use on microcomputers. It has a simplified syntax with no awkward terminators

PROMAL 2.0 FEATURES

COMPILED LANGUAGE

- Structured indentation syntax
- No line numbers or terminators
- Long variable names (31 characters)
- Global, Local, & Argument variables
- Byte, Word, Integer & Real data types
- Decimal or Hex number types
- Functions & Procedures with passed arguments
- Predefined DATA of any type
- Multi-Dimensional Arrays (any type)
- Strings & pointers
- Control Statements: IF-ELSE, WHILE, FOR, CHOOSE, REPEAT-UNTIL, BREAK, NEXT, INCLUDE, ESCAPE, REFUGE
- Bit-operators, shifts, type casts
- Variables at any memory location
- Simple Machine Language interface
- Recursion supported
- Program chaining and overlays (IMPORT/EXPORT)
- Separate compilation of modules
- Load and run relocatable M/L programs
- Compile errors trapped for Editor

EXECUTIVE (APPLE II & C64 Only)

- Command driven, with line editing
- Multiple user programs in memory at once
- Function key definitions
- Program abort and pause
- Prior command recall
- I/O Re-direction & batch jobs
- "DOS"-like commands: COPY, RENAME, DELETE, display FILES, TYPE, HELP, etc.
- Memory MAP, SET, and display commands

EDITOR

- Full-screen, cursor driven
- Function key controlled
- Line insert, delete, search
- String search and replace
- Block copy, move, delete & file read/write operations
- Auto indent, undent support

LIBRARY

- 50 Resident Machine-language commands
- Call by name with arguments
- String handling (9 routines)
- Re-directable I/O (STDIN & STDOUT)
- Formatted numeric output
- Decimal & Hexadecimal I/O
- Block fill/move/read/write
- Cursor control & line editing
- Data type conversion
- Random number function
- Real function support (in PROMAL): ABS, ATAN, COS, EXP, LOG, LOG10, POWER, SIN, SQRT, TAN
- Modem device support & much more

like ";" or "}" and indentation is part of the syntax, so structuring your code is natural and easy. Just compare PROMAL with BASIC in this example:

Equivalent Program Segments

PROMAL	BASIC
REPEAT	11910 REM
PROMPT AT 5,24, "Add/Chg/Quit?"	11920 CL = 5:LN = 24:PRS = "Add/Chg/Quit?"
IF Reply = "A"	11925 GOSUB 9490:REM GET REPLY
ADD Item	11930 IF PRS<>"A" THEN 11950
New_Items = New_Items + 1	11940 IF PRS<>"A" THEN 11950
ELSE IF Reply = "C"	11945 NI = NI + 1:GOTO 11920
CHANGE Item	11950 IF PRS<>"C" THEN 11970
UNTIL Reply = "Q"	11960 IF PRS<>"C" THEN 11970
	11970 IF PRS<>"Q" THEN 11920

PROMAL is readable and understandable. You see the logic from the structure. And PROMAL lets you call procedures by name—so no more GOSUBs. But there's more.

Slick Editor

Editing your source is a snap with the specially-designed and integrated **full-screen Editor**—it not only helps you structure your program, it **even finds compilation errors**—automatically.

Quick Compiler

The compiler is a **lightning-fast**, one-pass, recursive descent design. On the IBM PC it crunches source to object at **2000 lines per minute**, and it's equally impressive on the Apple and C64. And your PROMAL source code is **portable from machine to machine**. That means your source can be used on all PROMAL target machines.

Run-Time Speed Demon

PROMAL blows away Apple II and C64 languages from BASIC and PASCAL to FORTH. (Send \$3 for a copy of our full benchmark report.) It's **2000% faster** than BASIC. And on a normal IBM PC, the native 8088 code from **PROMAL beat Turbo Pascal 3.0** by 10% on the standard sieve benchmark!

DOS For Those Without

If you don't have a real "DOS," then PROMAL gives you a **true operating system environment** with the built-in operating system Executive. (See box.)

Outside Opinion

Naturally we're enthusiastic about PROMAL, but here's what other programmers are saying:

"Excellent... an ideal development system. ... Well done indeed!"

M. T. V.
Naperville, Ill.

"I am... so amazed by PROMAL... I cannot believe the high degree of excellence of this entire package."

C. P., Ph.D.
Ridgeway, New York

"I don't know that I've ever seen a [system] as thoughtfully designed and as skillfully executed as PROMAL. Its logic and ease of programming are truly remarkable. Its speed of execution is phenomenal... congratulations."

E. C. R.
Alexandria, VA

Safety In Numbers

SMA, Inc. has been satisfying customers (over 100,000) since 1982 with innovative microcomputer products. Now you can join our thousands of satisfied PROMAL users, by trying it today.

Try It For 30 Days On Us

Send us some bucks and we'll send you PROMAL on trial for 30 days. **If for any reason whatsoever you are not satisfied, just send it back** for a quick refund of your purchase price. No questions asked. No risk.

How To Order.

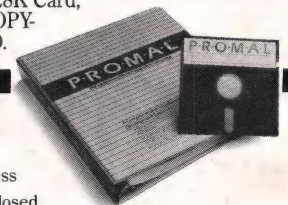
Call TOLL-FREE to order with your credit card or use the handy order form below to send in your check or money-order for your 30-day trial. Don't wait, you deserve the power of PROMAL today!

1-800-762-7874

In NC: 919-878-3600

SMA Systems Management Associates, Inc.
3325 Executive Drive, Dept. D-2
Raleigh, North Carolina 27609

PROMAL runs on IBM PC/PCjr with 192K, Commodore 64/128, APPLE IIc, or APPLE IIe with 80 Col. 128K Card, and is NOT COPY-PROTECTED.



Order Form for PROMAL 30-Day Trial!

My system is (check one)

- ☐ IBM PC/100% compatibles ☐ APPLE IIc/IIe
☐ COMMODORE 64/128

Please RUSH me:

- ☐ PROMAL Developer's System—Compiler, Editor, Library, Demo disk, 280-page manual, (Plus Executive for Apple and C-64) and stand-alone program generation (no royalties).

\$99.95 + 5.00 Shipping & Handling

- ☐ End-User System for Apple IIc/IIe and Commodore 64/128—all features of Developer's Version except stand-alone program generation (Executive needed for program execution)

\$49.95 + 5.00 Shipping & Handling

- ☐ Graphics ToolBox (Apple/C64 only)—20 routines for hi-res graphics: windows, clipping, text-on-graphics using scaled, rotated, user-defined fonts.

\$29.95 + 2.50 Shipping & Handling



Please charge my

- ☐ Visa
☐ MasterCard
☐ American Express

☐ My check is enclosed

Card Number _____

Signature _____ Exp. Date _____

Name _____

Address _____

City _____

State _____ Zip _____

NC residents add 4½% sales tax.
Foreign orders add \$20.00 additional shipping.

Listing One

```

000C 0000      prev:  dw      0
;
;         if      alone
rmvflg: db      0          ;keep this in memory
;         else
000E FF        rmvflg: db      0ffh      ;remove when main program ends
;         endif
;
000F 00        nonbnk: db      0          ;banked system
0010 42414E4B53  db      'BANKSWAP'
0018 00        loader: db      0
0019 0000      db      0,0
; *** note: If BANKSWAP is to be attached directly to SID.COM
; or DDT.COM then make rmvflg 0ffh to force removal
;
bankexam:
001B 210000      lxi      h,0          ;get stack pointer
001E 39          dad      sp          ;...and hold it for return
001F 220F04      shld     holdsp      ;...reset SP to a location
0022 310804      lxi      sp,locstk    ;...in common memory
;
; Main loop--exit by Quit or Remove command
bankex2:
0025 CD2E00      call     prompt      ;this returns address of sub
0028 CD6D00      call     doit        ;do subroutine addressed by HL
002B C32500      jmp      bankex2
;
002E 111F02      prompt: lxi      d,menu ;get ready for menu
0031 3A9103      lda      quietflag
0034 B7          ora      a          ;suppress menu?
0035 CA3B00      jz       prmp2
0038 11F402      prmp1: lxi      d,query ;set for prompt only
003B 0E09        prmp2: mvi      c,printf
003D CD0500      call     bdos
; Get a character
0040 0E01        mvi      c,conin
0042 CD0500      call     bdos
0045 CD7001      call     crlf
; make upper case, reject non-alpha
0048 E65F        ani      5fh
004A FE41        cpi      'A'
004C DA3800      jc       prmp1      ;ask again if invalid
004F FE5B        cpi      'Z'+1
0051 D23800      jnc       prmp1      ;ask again if invalid
; find match in alphtbl
0054 010700      lxi      b,altblen    ;length of table
0057 211102      lxi      h,alphtbl+altblen ;work back
005A BE          try:  cmp      m
005B CA6300      jz       match
005E 2B          dcx      h
005F 0D          dcr      c          ;count down
0060 C25A00      jnz      try
; if c= 0 no match found. Now form address
0063 211102      match: lxi      h,addrtbl
0066 09          dad      b          ;add offset
0067 09          dad      b          ;again, 2 bytes per table entry
; get the command address
0068 7E          mov      a,m
0069 23          inx      h
006A 66          mov      h,m
006B 6F          mov      l,a
006C C9          ret              ;HL has action address
;
006D E9          doit:  pchl          ;call here to use action address
;
getbank:
006E 2A9C03      lhld     b0start ;setup addresses

```

(Continued on page 42)

We've just invented an Algorithm Developer.

It just happens to be the best programmable software calculator on the market today.

PRO/SCI™ 99 bucks and a phone call.

Consider its performance.

- Programmable in simple BASIC—like statements
- Runs in *both* foreground and background modes
- Set up: Requires one key stroke
- Runs in background mode with most popular products (e.g. LOTUS, WordStar, dBase)
- Selects and moves rows of numbers *from* foreground document to calculator for any computation (e.g., standard deviation for 22 rows of numbers requires six key strokes)
- Returns results (at user's option) of all calculations to the foreground document
- Single-Step mode aids in algorithm de-bug

Functions

sin, cos, tan, asin, ln, exp, log, sqrt, abs, int, frac, round, mean, std, var, perm, comb, mod, rand, sum, min, max, integrate

Operators

arithmetic: +, -, *, /
 logical: #AND#, #OR#, #NOT#
 relational: >, <, >=, <=, ==, <>
 exponentiation: **, ^
 factorial: !

- Spreadsheet comparison
 - Greater range of formula capabilities
 - Access requires only key stroke
 - Transfers numbers to calculator without retyping
- Full formula line editing
- Ten formulas per formula file
- Unlimited number of formula files
- 8087 version available
- All calculations double precision
- Includes Accountant's style calculator with HEX, OCT, and BIN!
- Runs with SIDEKICK — in all modes

Programming Statements

Conditional and absolute looping and branching
 Indexing by variable increment and test
 Formula con-catenation (to 1000 columns)
 Printing statements for line printer tabulations
 Debug mode — single-step and display of all variables
 Displayed decimal point control

Constants

pi, e, c, K, h, q, R, N₀, G, g

```

amort.CFG
6-amrt
months=months+1;pr=((pmt*12/int)/(1+int/12)
^(mths+1))*[1-(1+int/12)];@next
7-amrti
in=-pmt-((pmt*12/int)/(1+int/12)^(mths+1))
*[1-(1+int/12)];@next
8-sumr
prinpd=prinpd+pr;intpd=intpd+in;mths=mths-1;
@ifqt (mths==168);@goto(6)
A sample formula listing
    
```

Send me **PRO/SCI™** for \$99 and include 8087 Version at NO EXTRA CHARGE.
 (Plus \$5.00 shipping & handling)

☐ American Express ☐ VISA
☐ Money Order ☐ MasterCard
☐ Check
 Card No. _____
 Expiration Date _____
 Signature _____

Requires IBM PC or true compatible with 128K RAM and one disk drive and MS-DOS 2.1 or higher

NAME _____
 ADDRESS _____
 CITY/STATE/ZIP _____
 TELEPHONE _____

California residents add 7% sales tax. Outside U.S.A. add \$15.00. (If outside of U.S.A. payment must be by bank draft payable in the U.S. and in U.S. dollars.) Sorry, no C.O.D.

SYMSOFT

P.O. Box 4477
 Mtn. View, CA 94043

Orders Taken:
 Mon.-Fri., 8am-4pm

800/227-6703
 OUTSIDE CALIFORNIA

800/632-7979
 INSIDE CALIFORNIA

Listing One

```

0071 220A04      shld    source
0074 2A9E03      lhld    blstart
0077 220C04      shld    dest
007A 3AA103      lda     length+1      ;low byte ignored
007D 320E04      sta     count

; page by page take a chunk of bank 0 to buffer, then move
; to bank 1 destination
get2:
0080 F3          di          ;be sure no interrupts
0081 3E00        mvi        a,0
0083 2A0804      lhld    selmv
0086 CD6D00      call    doit
0089 CDF400      call    getbuf
008C 3E01        mvi        a,1
008E 2A0804      lhld    selmv      ;point to BIOS selmem routine
0091 CD6D00      call    doit
0094 FB          ei          ;interrupts safe again
0095 CDFD00      call    putbuf
0098 110001      lxi        d,buflen
009B 2A0A04      lhld    source
009E 19          dad        d
009F 220A04      shld    source
00A2 2A0C04      lhld    dest
00A5 19          dad        d
00A6 220C04      shld    dest
00A9 210E04      lxi        h,count ;repeat for required # of pages
00AC 35          dcr        m
00AD C28000      jnz      get2
00B0 C9          ret

;
putbank:
00B1 2A9E03      lhld    blstart
00B4 220A04      shld    source
00B7 2A9C03      lhld    b0start
00BA 220C04      shld    dest
00BD 3AA103      lda     length+1      ;low byte ignored
00C0 320E04      sta     count

; page by page, move bank 1 data to buffer, then move it
; to bank 0
put2:
00C3 CDF400      call    getbuf
00C6 F3          di          ;be sure no interrupts
00C7 3E00        mvi        a,0
00C9 2A0804      lhld    selmv
00CC CD6D00      call    doit
00CF CDFD00      call    putbuf
00D2 3E01        mvi        a,1
00D4 2A0804      lhld    selmv
00D7 CD6D00      call    doit
00DA FB          ei          ;interrupts safe again
00DB 110001      lxi        d,buflen
00DE 2A0A04      lhld    source
00E1 19          dad        d
00E2 220A04      shld    source
00E5 2A0C04      lhld    dest
00E8 19          dad        d
00E9 220C04      shld    dest
00EC 210E04      lxi        h,count
00EF 35          dcr        m
00F0 C2C300      jnz      put2
00F3 C9          ret

; source to buffer
getbuf:
00F4 2A0A04      lhld    source
00F7 111104      lxi        d,buffer
00FA C30401      jmp      pbl

; buffer to dest
putbuf:

```

(Continued on page 44)

C Programmers:

Consider 104 Ways To Be More Productive

If you find and choose the right development software, you can: cut development effort, make impractical projects feasible, and eliminate unproductive, frustrating aspects of programming.

Confused? We'll help you sort thru the huge number of alternatives. Call for comparisons or information.

Learn C Programming Only \$95

"Introducing C" Interpreter

Computer Innovations has done it again! This interactive implementation is combined with a full screen editor and a thorough, self-paced manual.

You can develop programs faster by getting immediate feedback. Programs will start instantly upon your command. There is no need to wait "for compile and link."

Introducing C includes demo programs, powerful C language interpreter, complete C function library, full screen editor, color graphics, and C language compatibility. PCDOS \$95

Add Communications Features to Your Programs Greenleaf Comm Library

Greenleaf now enables you to communicate with remote systems or databases with an asynchronous communications library for C.

Individual transmission and reception ring buffers combine with an interrupt driven system. This eliminates the extra function of separately calling up the communications program.

Included are 3 library/object files, 220 functions; 230 page manual, complete source code, library tailor-made to suit compiler and memory. Hayes-compatible modem commands, and a complete sample file transfer program. MSDOS \$169

Shorten Development Time, Cut Frustrations BRIEF, The Programmer's Editor

Compile within BRIEF; use autoindent; "templates", C specific error support... use windows, UNDO, and multiple large files.

But edit YOUR WAY.

Every feature you'd expect and more are integrated elegantly — and it can be modified by you.

You deserve:

"...the best text editor you can buy." — John Dvorak, InforWorld, 7/8/85
"...the best code editor..." John Irwin, Data Based Advisor, 8/85

PCDOS \$Cali

Fast File Access with Source C-Index +

C-Index + contains a high performance ISAM, balanced B + Tree indexing system with *source* and *variable length* fields. The result is a complete data storage system to eliminate tedious programming and add efficient performance to your programs.

Features include random and sequential data access, virtual memory buffering, and multiple key indexes.

With *no royalties* for programs you distribute, full source code, and variable length fields C-Index + fits what you are likely to need.

Save time and enhance your programs with C-Index +. MSDOS \$375

We carry 27 C Compilers, 4 C Interpreters, 49 Support Libraries, 5 C source debuggers, and 19 other C Add-ons for programming with MSDOS, Macintosh, or CP/M — more than 104 products, really. Here are some of the best products available:

First Aid for C Programs C ToolSet

Save time and frustration when analyzing and manipulating C programs.

DIFF and CMP - for "intelligent" file comparisons.

XREF - cross references variables by function and line.

C Flow Chart - shows what functions call each other.

C Beautifier - make source more readable.

GREP - search for patterns.

There are several other programs for converting and printing programs.

Portable. Full source code.

CPM, MSDOS \$135

SORT/MERGE Files for Clean, Fast Maintenance with OPT-TECH SORT

Performance should not suffer with DOS or other "free" sorts. ISAMs alone are slow when 10% or even less is changed/added.

OPT-TECH includes:

- CALLable and Standalone use
- C, ASM, BAS, PAS, FTN, COBOL
- Variable and fixed length
- 1 to 9 fields to sort/merge
- Autoselect of RAM or disk
- Options: dBASE, Btrieve files
- 1 to 10 files Input
- No software max for # Records
- All common field types
- By pass headers, limit sort
- Inplace sort option
- Output = Record or keys

Try what you're using on an XT: 1,000 128 byte records, 10 byte key in 33 seconds. MSDOS. \$90.

Get File Access with TIGHTER Control than an ISAM or a Sort. db_VISTA Data Management

6 files updating at once, entering data with validation...with no delays! Multiuser access at the same time. db_VISTA can do it.

db_VISTA makes practical sophisticated applications that cannot be handled with reasonable performance by a "relational" or "flat file" system. But, if you want a simple ISAM, it is included too.

Full source, no royalties and "normal" indexed file management are part of db_VISTA. Get more for the price of only an ISAM.

You can minimize data stored and access records even faster and more logically than just using indexes. Example: address and transaction data should not require redundant storage of customer names or numbers. Use pointers. Related data fields point to other related groups — the "network model" of data.

Use db_VISTA as a "normal ISAM" or save programming time, access time and file size. Lattice, C86, Williams, Desmet, Microsoft C.

MSDOS Multiuser source \$945, Object \$445
Single user source \$445, Object \$275

Call for details, comparisons, or for our "C Extras Packet" with over 50 pages of information about C support products.

THE PROGRAMMER'S SHOP

The programmer's complete source for software, services and answers

128-DC Rockland Street, Hanover, MA 02339 (617) 826-7531 (800) 421-8006

Ask about COD and PO's. All formats available. Prices subject to change. Names of products and companies are generally their trademarks.

Circle no. 141 on reader service card.

Listing One

```

00FD 2A0C04      lhld    dest
0100 111104      lxi     d,buffer
0103 EB          xchg
; common code for getbuf and putbuf
0104 010001      pbl:    lxi     b,buflen
; move (BC) bytes from (HL) to (DE) (could use BIOS move
; routine for this)
0107 7E          move:   mov     a,m
0108 12          stax    d
0109 23          inx     h
010A 13          inx     d
010B 0B          dcx     b
010C 78          mov     a,b
010D B1          ora     c
010E C20701      jnz     move
0111 C9          ret

;
; Go back to SID/DDT
0112 116403      quit:   lxi     d,qmsg ;say how to get back
0115 0E09        mvi     c,printf
0117 CD0500      call    bdos
011A 2A0F04      lhld    holdsp ;restore stack pointer
011D F9          sphl
011E FF          rst     7 ;back to DDT or SID

;
; if alone
; Set for removal on termination of SID or DDT
remove: lxi     h,rmvflg ;set the remove flag
mvi     m,0ffh ;...in the RSX prefix
lxi     h,rstv ;then wipe out the entry
mvi     m,0ffh ;...JMP with an RST 7
rst     7 ;leave via the debugger
endif ;alone

;
; Set up addresses for move, also set up length
adset: lxi     d,b0id ;tell current bank 0 addr
mvi     c,printf
call    bdos
0127 2A9C03      lhld    b0start ;get the address
012A CDDA01      call    addro ;and print it
012D 219C03      lxi     h,b0start
0130 CD8101      call    update ;enter hex to (HL)
0133 112503      lxi     d,blid ;do it again for bank 1
0136 0E09        mvi     c,printf
0138 CD0500      call    bdos ;tell
013B 2A9E03      lhld    blstart
013E CDDA01      call    addro
0141 219E03      lxi     h,blstart
0144 CD8101      call    update ;get new address, if any
; Can fall through from adset or enter directly here to set
; length of block moved
0147 113303      lnset:  lxi     d,lnmsg ;tell current length
014A 0E09        mvi     c,printf
014C CD0500      call    bdos
014F 2AA003      lhld    length
0152 CDDA01      call    addro
0155 21A003      lxi     h,length
0158 CD8101      call    update ;offer to change it
015B CD7001      call    crlf
015E C9          ret

;
; Toggle menu off/on
015F 3A9103      xpert:  lda     quietflag
0162 2F          cma
0163 329103      sta     quietflag ;toggle
0166 C9          ret

;
; Get here on invalid command

```

(Continued on page 46)

VEDIT PLUS

NEW FEATURES MAKE IT BETTER THAN EVER

VEDIT is the most often reviewed and highly acclaimed text editor ever. And no wonder - we have listened to and supported our users for almost six years and have continually improved our products by incorporating hundreds of their suggestions. Now we have a revolutionary new version of VEDIT PLUS with all the program development features you've asked for.

The unique on-line interactive help is a first - it can be easily expanded or edited. You can create your own on-line help with menus and sub-menus for compilers, style guides, even non-computer subjects.

NEW FEATURES

- Memory management supports up to 1 MByte
- MS-DOS, PC DOS pathname support
- Interactive on-line help
- Paragraph justification
- Create your own edit functions with macros
- Auto-execute macros as application programs
- Easy to use installation/customization program for keyboard assignment, screen attributes, etc.

The supplied macros are menu driven and immediately usable. Sort a mailing list. Perform a full screen file comparison and update one file from the other - merge the work done by two programmers on the same file. Replace the command prompt with a menu for new users. And an exceptional 8080 to 8086 translator is an inexpensive option.

We have a Surprise for You.

Your own editing macros are easy to write and will automate otherwise tedious chores. You can use a macro the instant you type it or load it from disk - there is no time wasted compiling or linking. Macros are accessible from function keys - allowing you to create your own editing functions.

- Simultaneously edit up to 36 files of unlimited size
- Unlimited file handling - merge, split, extract, view, print and more
- Search with wild cards and patterns
- On-line integer calculator
- Complete macro programming language including:
 - IF-THEN-ELSE and looping
 - User prompts and input
 - Algebraic expressions with variables
 - Complete TECO capability
- Free macros

VEDIT PLUS isn't only for programming. The word processing features include word wrap, paragraph formatting with justification, printing and horizontal scrolling for spreadsheets. Converts WordStar files and edits dBASE™ source files. Add V-PRINT for fancier features such as automatic index generation and printer font support (incl. laser printers). Then add V-SPELL, the ultimate spelling corrector, and you'll have a word processing system that can do virtually everything you need.

As a programming or writing professional, VEDIT PLUS gives you refinement, power and flexibility you will never outgrow.

VEDIT PLUS supports virtually every MS-DOS, PC DOS, CP/M-86 and CP/M computer and is a favorite among OEMs. VEDIT PLUS is not copy protected. Order direct or from any of our over 400 dealers. \$225. (Discounts available for current VEDIT users.) Site licensing and educational discounts.

CompuView

1955 PAULINE BLVD., ANN ARBOR, MICHIGAN 48103
(313) 996-1299, TELEX - 701821

Listing One

```

0167 11FA02    na:      lxi      d,namsg      ;say can't do it
016A 0E09      mvi      c,printf
016C CD0500    call     bdos
016F C9        ret

;
0170 C5        crlf:    push     b
0171 D5        push     d
0172 E5        push     h
0173 F5        push     psw
0174 111403    lxi      d,crlfstring
0177 0E09      mvi      c,printf
0179 CD0500    call     bdos
017C F1        pop      psw
017D E1        pop      h
017E D1        pop      d
017F C1        pop      b
0180 C9        ret

;
; get string, do nothing if null, else convert, store at (HL)
update:
0181 E5        push     h      ;save the location
; loop back here if input is not valid
updl:  lxi      h,0          ;initial buffer
0182 210000    shld     inbuf+1
0185 229303    lxi      d,chquery      ;say what's up
0188 114203    mvi      c,printf
018B 0E09      call     bdos
018D CD0500    call     bdos
0190 119203    lxi      d,inbuf
0193 0E0A      mvi      c,rdbuf      ;read console until <ret>
0195 CD0500    call     bdos
0198 CD7001    call     crlf
019B 3A9303    lda      inbuf+1      ;get length of hex string
019E B7        ora      a          ;check for 0 length input
019F C2A401    jnz      convert
; null string, go back
01A2 E1        pop      h          ;retrieve value at entry
01A3 C9        ret

;
; Convert the hex string in the buffer to binary
convert:
01A4 210000    lxi      h,0          ;start with a zero
01A7 47        mov      b,a          ;hold length in B
01A8 119403    lxi      d,inbuf+2
01AB 1A        conv2:  ldax     d      ;get first (or next) char
01AC 13        inx      d
01AD FE60      cpi      60h
01AF DAB401    jc       conv3
01B2 E65F      ani      5fh          ;make lower case if necessary
01B4 D630      conv3:  sui      '0'
01B6 FA8201    jm       updl        ;must be valid hex, 0..9, A..F
01B9 FE0A      cpi      0ah
01BB DACA01    jc       num        ;jump if a good numeric
01BE D607      sui      7
01C0 FE0A      cpi      0ah
01C2 DA8201    jc       updl        ;error if not good alpha
01C5 FE10      cpi      10h
01C7 D28201    jnc      updl        ;error if not good alpha
01CA 29        num:    dad      h      ;multiply current val by 16
01CB 29        dad      h
01CC 29        dad      h
01CD 29        dad      h
01CE 85        add      l          ;add new least significant digit
01CF 6F        mov      l,a
01D0 05        dcr      b          ;countdown the digits
01D1 C2AB01    jnz      conv2
01D4 EB        xchg
01D5 E1        pop      h          ;HL at entry says where to it

```

(Continued on page 48)

More Power Than You Thought Possible

Arity offers the first serious implementation of Prolog for IBM personal computers. Arity/Prolog is a powerful, highly optimized, and extended version of the logic programming language Prolog. Imagine building software applications with a language that solves problems through deduction and logical inference. The task of creating complex programs is much faster and easier, resulting in lower development costs. Arity/Prolog is now in use in a wide range of applications in industry, business, research, and education. The solution—the *Arity/Prolog Interpreter*:

- Source level debugger
- Virtual databases, each with a workspace of 16 megabytes
- Floating-point arithmetic
- String support for efficient text handling



- Interface to assembly language and 'C'
- Text screen manipulation
- Integrated programming shell to MSDOS
- Comprehensive set of evaluable predicates
- Definite clause grammar support

Arity/Prolog Interpreter \$495.00

Arity also offers the *Arity/Prolog Compiler and Interpreter*, a sophisticated development environment for building AI applications. Essential for producing fast, serious production code.

Arity/Prolog Compiler and Interpreter \$1950.00

The *Arity/Prolog Demo Disk* is available for \$19.95. ■ Arity/Prolog products run on the IBM PC, XT, AT, and all IBM compatibles. ■ To order, call (617) 371-2422 or use the order form below.

arity corporation 358 Baker Avenue, Concord, MA 01742

Name _____

Organization _____

Address _____

☐ Enclosed is a check or money order to Arity Corporation

☐ Please bill my

☐ Mastercard ☐ Visa ☐ American Express

Account #

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Valid ____/____/____ to ____/____/____
signature _____

Quantity	Product	Unit Price	Total Price
	Arity/Prolog Compiler & Interpreter	\$1950.00	
	Arity/Prolog Interpreter	\$ 495.00	
	Arity/Prolog Demo Disk	\$ 19.95	
Subtotal			
MA residents add 5% sales tax			
Total Amount			

☐ Please send me more information about Arity and Arity/Prolog

arity 358 Baker Avenue, Concord, MA 01742

M-AD-05

Listing One

```

01D6 73          mov     m,e
01D7 23          inx     h
01D8 72          mov     m,d
01D9 C9          ret

;
; Print HL as hex
addro:
01DA D5          push    d
01DB E5          push    h
01DC EB          xchg
01DD 215C03      lxi     h,hexstr          ;where to build string
01DE 7A          mov     a,d
01E1 CDF301      call    byte          ;get A as 2 ASCII chars at (HL)
01E4 7B          mov     a,e
01E5 CDF301      call    byte          ;again, low byte
01E8 115C03      lxi     d,hexstr
01EB 0E09        mvi     c,printf
01ED CD0500      call    bdos          ;print it
01F0 E1          pop     h
01F1 D1          pop     d
01F2 C9          ret

;
; Convert byte to hex ASCII chars, put at (HL)
byte: 01F3 F5      push    psw
01F4 1F          rar          ;get high nybble
01F5 1F          rar
01F6 1F          rar
01F7 1F          rar
01F8 CDFC01      call    nybble
01FB F1          pop     psw          ;fall through for low nybble
; nybble makes 1 char, advances output pointer
nybble: 01FC E60F      ani     0fh
01FE C630        adi     '0'
0200 FE3A        cpi     3ah
0202 DA0702      jc      nput
0205 C607        adi     7
nput: 0207 77      mov     m,a
0208 23          inx     h
0209 C9          ret

;
; Acceptable command inputs go in this table
alphtbl:
020A 00          db      0          ;dummy for no match
020B 41          db      'A'
020C 47          db      'G'
020D 4C          db      'L'
020E 50          db      'P'
020F 51          db      'Q'

;
;
if alone
db 'R'
endif ;alone

;
0210 58          db      'X'
0007 =          altblen: equ     $-alphtbl
; addresses of action routines, same order as alphtbl
addrtbl:
0211 6701        dw      na          ;not available
0213 1F01        dw      adset       ;address set
0215 6E00        dw      getbank
0217 4701        dw      lnset       ;length set
0219 B100        dw      putbank
021B 1201        dw      quit

;
if alone
dw remove
endif

;

```

(Continued on page 50)

THE PROGRAMMER'S SHOP™

helps save time, money and cut frustrations. Compare, evaluate, and find products.

SERVICES

- Programmer's Referral List
- Compare Products
- Help find a Publisher
- Evaluation Literature FREE
- BULLETIN BOARD - 7PM to 7AM 617-826-4086
- Dealer's Inquire
- Newsletter
- Rush Order
- Over 700 products

SERVICE: FREE NEWSLETTER

Software development and AI on micros: trends, forecasts, controversies, innovations, and techniques. Plus announcement of 80 NEW tools. CALL for "Newsletter Packet."

RECENT DISCOVERY

Periscope II symbolic debugger - some source level, 2 screens, interrupt switch \$129

AI - Expert System Dev't

ExpertEASE - Inductive. PC DOS Call
EXSYS - All RAM, Probability. Why.
Trees, Solid, files, popular PC DOS \$359
INSIGHT 1 - Probabilities, required
thresholds, menus, fast PC DOS \$ 95
INSIGHT 2 - adds backward, forward, par-
titions, dB2, lang. access. PC DOS \$449
Others: APES (\$359), Advisor (\$949),
ES Construction (\$100), ESP (\$845),
Expert Choice (\$449), more.

AI - LISP

List Our

GC LISP - "Common", rich.
Interpreter - Interactive Tutorial 495 Call
LARGE Model - 2 to 15 meg. 695 649
Compiler and Interpreter 1190 1045
IQ LISP - MacLISP and InterLISP.
Well-liked, 1,000K. PC DOS \$149
TLC LISP - "LISP-Machine" - like, all
RAM, classes, turtle graphics, 8087,
compiler. CPM-86, MSDOS \$235
WALTZ LISP - "FRANZ LISP" - like, big
nums, debug, CPM-80, MSDOS \$149
Others: ExperLISP (\$439), TransLisp-PC
(\$75), BYSO (\$125), MuLISP-86 (\$199)

AI - PROLOG

ARITY PROLOG - full, debug, ASM, C,
virtual. Compiler \$1950 MSDOS \$495
MPROLOG - Rich syntax, editor, segment
work space, portable. PC DOS \$725
Prolog-86 - Learn Fast. Standard,
tutorials, samples MSDOS Call
Others: Prolog-1 (\$359), Prolog-2 (\$1895),
Prolog V (\$65), V + (\$95), MicroProlog
(\$229), Prof. MicroProlog (\$359).

Editors for Programming

BRIEF Programmer's Editor - undo,
windows, reconfigure PC DOS Call
C Screen with source 80/86 \$ 75
EMACS by UniPress - powerful,
multifile, windows, DOS, MLISP.
programming. Source: \$949 MSDOS
\$299
Entry System for C - Bellesoft PC DOS \$325
FirsTime by Spruce - Improve
productivity. Syntax directed for Turbo
(\$69), Pascal (\$229), or C (\$239)
PMATE - power, multitask 80/86 \$159
VEDIT - well liked PC DOS \$119
XTC - multitasking PC DOS \$ 95

Feature

MultiHALO - full 2.0, all graphics
boards, printers, comprehensive.
For Turbo (\$95), for GC LISP
(\$250), Others (\$195).

C Language - Compilers

BDS C - solid value, fast CPM80 \$125
C86 by CI - 8087, reliable MSDOS Call
Hippo Level 2 (\$375), Lev. 1 MAC \$109
Lattice C - from Lifeboat MSDOS \$289
from Lattice MSDOS \$339
Consulair Mac C w/toolkit MAC \$299
Megamax - tight, full MAC \$239
Microsoft C 3.0 - new, tight. MSDOS \$259
Q/C 88 by CodeWorks - **Compiler source**,
decent code, cross/native MSDOS \$295
Williams - source debug. MSDOS \$399
Wizard C - Lattice C compatible,
full Sys. III, lint, fast. MSDOS \$399

C Language - Interpreters

C-terp by Gimpel - full K & R, .OBJ
and ASM, large progs. MSDOS \$249
INSTANT C - Source debug, Edit to
Run-3 seconds MSDOS \$399
INTRODUCING C - Interactive C to
learn fast, tutorial PC DOS \$115
RUN/C - improved MSDOS \$109

C Libraries - General

Blaise C Tools 1 (\$109), C Tools 2 \$89
C Food by Lattice - ask for source \$119
C*LIB by Vance \$125
C Utilities by Essential - 300 + \$149
Greenleaf Functions - portable & ASM \$149
Polytron - for Lattice, ASM source \$ 99
Software Horizons - Pack 1 \$129

C Libraries - Applications

COMMUNICATIONS: Asynch Mgr \$175
Greenleaf - full, respected \$149
Software Horizons - Pack 3 \$139
FILES: Btrieve - multilanguage \$199
C Index by Trio - full B + Tree,
vary length field, multi compiler
/File is object only \$ 89
/Pro is partial source \$179
/Plus is full source \$349
C Tree by Faircom - source, port. \$349
dbcISAM by Lattice for dB2 or 3 \$219
dbVISTA - full indexing, plus optional
record types, pointers. Network.
Object only - MS C, LAT, C86 \$179
Source - Single user \$459
Source - Multiuser \$929

Note: All prices subject to change without notice.
Mention this ad. Some prices are specials.
Ask about COD and PO's. All format's available.

Call for a catalog, literature, and solid value

800-421-8006

THE PROGRAMMER'S SHOP™

128-D Rockland Street, Hanover, MA 02339

Mass: 800-442-8070 or 617-826-7531 1085

C Support - Systems

C Debug - Source debuggers - by
Complete Soft (\$269), MSD (\$149).
CPRINT - by ENSCO, thorough \$ 50
C Sharp - well supported, Source,
realtime, tasks MSDOS \$600
C Sprite Debugger by Lattice \$149
C ToolSet - DIFF, xref, source \$135
Faster C - replace LINK \$ 95
PC Lint - full C program checking
and big, small model. All C's \$119

Low Cost Languages

C/86 by Software Toolworks -Quick \$ 65
Introducing C - Step by step training \$109
Let's C by Mark Williams \$ 59
TransLisp-PC - "Common Lisp", tutorial,
graphics, 230 functions, samples \$ 75
Modula 2 by ITC - Windows, tight \$ 80
PC/Forth by LMI - '83, DOS, ASM \$129
Prolog-86 - enhanced, DOS, Edit Call
Quick BASIC by Microsoft - Compile
BASICA, Link \$ 79
Snobol4+ - by Catspaw - Strings \$ 85

TURBO PASCAL and SUPPORT

BORLAND: Turbo 3.0 \$ 49
3.0 with 8087 or BCD \$ 79
3.0 with 8087 and BCD \$ 85
Turbo Graphix - graphs, windows \$ 39
Turbo Toolbox \$ 39
Turbo Tutor \$ 29
TURBO . . . Asynch by Blaise, full \$ 89
Power Tools by Blaise - library \$ 89
Power Utilities - profiler, pp \$ 89
MetaWindow by Metagraphics \$ 49
OTHERS: FirsTime (\$69), Screen
Sculptor (\$99), Pascal Pac (\$100),
Tidy (\$45), Multi Halo (\$95).

Fortran & Supporting

Forlib+ by Alpha - graph, comm. \$ 59
MACFortran by Microsoft - full '77 \$239
MicroSubMath - Trig, Vessel \$250
PolyFortran - xref, pp, screen \$149
RM Fortran - enhanced "IBM Ftn" \$429
Scientific Subroutines - Matrix \$149
Strings and Things - registers, shell \$ 59

MultiLanguage Support

Btrieve/N (\$469), single user \$199
Codesifter - executive profiler \$109
DoubleDOS - nice, two task \$79
LMK Make by Lattice \$159
MultiLink - multitasking \$249
Panel - Screens, windows \$239
PFinish - Profile by line, routine \$299
PLink-86 - 32 levels, overlays \$289
PolyLibrarian - Manage .OBJS \$ 89
TexSys - Source code control \$ 89

Listing One

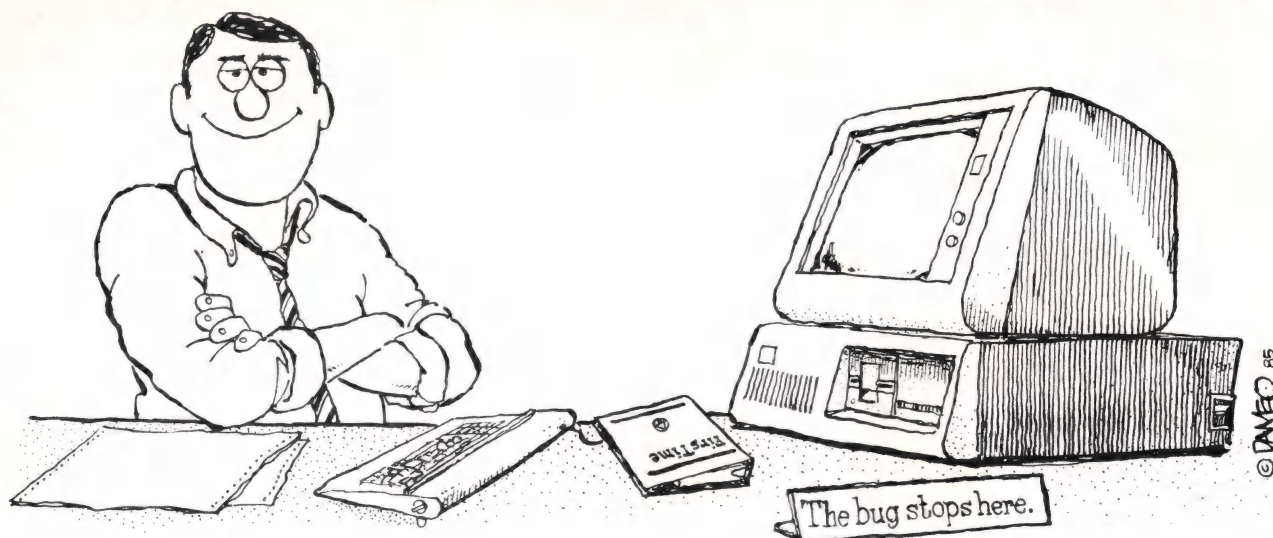
```

021D 5F01          dw      xpert      ;expert mode, no menu
;
021F 42616E6B73menu: db      'Bankswap 1.0 by A. S. Woodhull
10/20/83',0dh,0ah
0248 66756E6374    db      'functions available:',0dh,0ah
025E 09412E2E2E    db      '      A...set move Addresses',0dh,0ah
0277 09472E2E2E    db      '      G...Get alternate bank',0dh,0ah
0290 094C2E2E2E    db      '      L...set move Length',0dh,0ah
02A6 09502E2E2E    db      '      P...Put alternate bank',0dh,0ah
02BF 09512E2E2E    db      '      Q...Quit to SID or DDT',0dh,0ah
;
; if alone
db      '      R...Remove BANKSWAP',0dh,0ah
endif
;
02D8 09582E2E2E    db      '      X...eXpert mode (no menu)',0dh,0ah
02F4 0D0A3F3F20query: db      0dh,0ah,'?? $'
02FA 2E2E2E6675namsg: db      '...function not available.'
;
; crlfstring:
0314 0D0A24        db      0dh,0ah,'$'
0317 42616E6B20b0id: db      'Bank 0 addr: $'
0325 42616E6B20blid: db      'Bank 1 addr: $'
0333 4C656E6774lnmsg: db      'Length is now $'
;
; chquery:
0342 4368616E67    db      'Change to? (CR to keep): $'
035C                ds      4
0360 480D0A24        db      'H',0dh,0ah,'$'
0364 52652D656Eqmsg: db      'Re-enter BANKSWAP from DDT or SID by "G28"'
038E 0D0A24        db      0dh,0ah,'$'
;
; quietflag:
0391 00            db      0          ;initialized off
;
0392 08            inbuf: db      8          ;max length of buffer
0393                ds      9
;
; default parameters: alter by Set and Length commands
039C 0001          b0start: dw      b0dft
039E 0001          blstart: dw      bldft
03A0 0010          length:  dw      movcnt
;
; One-time routine, on 1st BDOS call intercepted
install:
03A2 C5            push     b          ;keep everything as it was
03A3 D5            push     d          ;...so BDOS function can be
03A4 E5            push     h          ;...completed
03A5 F5            push     psw
; set up restart vector for re-entry
03A6 3EC3          mvi      a,0c3h    ;a JMP instruction
03A8 322800        sta      rstv
03AB 211B00        lxi      h,bankexam
03AE 222900        shld     rstv+1
; set up address of BIOS routine accessed directly
03B1 2A0100        lhld     biosv          ;find where BIOS is
03B4 114E00        lxi      d,selmem      ;...and add offset
03B7 19            dad      d
03B8 220804        shld     selmv
; then patch the RSX prefix to prevent reinstallation
03BB 2A0A00        lhld     next+1
03BE 220700        shld     start+1
; tell 'em we're ready
03C1 0E09          mvi      c,printf
03C3 11D003        lxi      d,msg
03C6 CD0900        call     next
03C9 F1            pop      psw          ;continue with the task that
03CA E1            pop      h          ;...was so rudely interrupted
03CB D1            pop      d
03CC C1            pop      b
03CD C30900        jmp      next

```

(Continued on page 52)

The First Idea-Processor For Programmers.



FirstTimeTM

Has features no other editor has.

- Fast program entry through single keystroke statement generators.
- Fast editing through syntax oriented cursor movements.
- Dramatically reduced debugging time through immediate syntax checking.
- Fast development through unique programmer oriented features.
- Automatic program formatter.

FirstTime is a True Syntax Directed Editor.

As the world's most advanced syntax-directed editor, FirstTime lets you work with ideas by taking care of the low-level syntax details of your program. For example, you can generate complete statement skeletons with one keystroke. Move the cursor from one procedure to the next with one keystroke. Type in code, and it's instantly formatted (you specify the rules). Type an error, and FirstTime warns you immediately. You can continue working if you wish. Later, you can use the search-for-error command to find the error and fix it.

FirstTime Has Thorough Error Checking.

FirstTime not only checks your syntax, but also semantics. FirstTime identifies:

- Undefined variables, types and constants.
- Assignment statements with type mismatches.
- Errors in include files and macro expansions.

To Order Call: (201) 741-8188 or write:

SPRUCE TECHNOLOGY CORPORATION



P.O. Box 7948
Shrewsbury, NJ 07701

FirstTime is a trademark of Spruce Technology Corporation • MS-DOS is a trademark of Microsoft Corporation • IBM is a trademark of International Business Machines Inc. • Turbo Pascal is a trademark of Borland International • dBase III is a trademark of Ashton-Tate.

FirstTime Lets You Work With Ideas.

The *Zoom command* gives you a top down view of your program logic.

The *View macro command* shows the expansions of a C macro while in the editor.

The *View include file command* instantly shows you the contents of an include file.

The *Transform command* allows you to change a statement to another similar statement, for example, a *FOR* to an equivalent *WHILE*.

The *Search for next error command* allows you to find errors throughout your program.

The *Cursor movement commands* let you traverse your program by logical elements, not just characters.

FirstTime Works With Existing Files.

FirstTime works with standard ASCII files, so you can edit any existing source files.

FirstTime for Turbo Pascal	\$ 74.95
FirstTime for dBase III	\$125.00
FirstTime for MS-Pascal	\$245.00
FirstTime for C	\$295.00

In Germany, Austria and Switzerland contact:
Markt & Technik Software Verlag
Munchen, W. Germany
(089) 4613-0

Listing One

```

;
03D0 42414E4B53imsg: db 'BANKSWAP loaded. To access from DDT or '
03F7 5349442074 db 'SID type "G28"'
0405 0D0A24 db 0dh,0ah,'$'

;
; reuse installation code area for stack space
locstk:
;
; uninitialized storage
0408 selmv: ds 2 ;used for BIOS call to selmem
040A source: ds 2 ;for moves to buffer
040C dest: ds 2 ;for moves from buffer
040E count: ds 1 ;blocks to move
040F holdsp: ds 2 ;stack pointer from DDT or SID

;
buffer:
0411 ds buflen
0511 end

```

End Listing One

Listing Two

```

; RSXMAKER.SUB assemble RSX and attach to existing COM file
; ASW 21 Oct 83
; Usage: A>SUBMIT RSXMAKER <comfile> <asmfile>
;
rmac $2
link $2 [op]
ren $2.rsx=$2.prl
gencom $1 $2

```

End Listing Two

Listing Three

Listing Three. An example of the use of BANKSWAP to allow examination of a portion of a custom disk drive routine installed in bank 0.
(Comments added in parentheses).

```

D>sid
BANKSWAP loaded. To access from DDT or SID type "G28"
CP/M 3 SID - Version 3.0
#g28 (go to BANKSWAP)
Bankswap 1.0 by A. S. Woodhull 10/20/83
functions available:
    A...set move Addresses
    G...Get alternate bank
    L...set move Length
    P...Put alternate bank
    Q...Quit to SID or DDT
    X...eXpert mode (no menu)

?? x (remove the menu)
?? a (set up range)
Bank 0 addr: 0100H
Change to? (CR to keep): 7000 (don't accept default)
Bank 1 addr: 0100H
Change to? (CR to keep): 7000
Length is now 1000H
Change to? (CR to keep): <CR> (length is OK)
?? g (get the data from bank 0)
?? q (back to SID)
Re-enter BANKSWAP from DDT or SID by "G28"
*d61E (we're back)
#d7c00 7c3f (examine the data from bank 0)
7C00: C3 09 7C C3 0F 7C C3 15 7C 21 94 0C C3 00 73 21 ..|...|!....s!
7C10: 74 0C C3 00 73 21 1B 0C C3 00 73 AE F1 0C A9 04 t...s!....s....
7C20: D0 2D AE F1 0C AC DF 0C B9 BA 0C 9D 8E C0 AD DE .-.....
7C30: 0C 4A 48 A9 00 2A 0D EF 0C 0D F1 0C A8 B9 80 C0 .JH..*.....
#^C (return to CP/M)
D>

```

End Listings

BRIEF™

"BRIEF, The Programmer's Editor, is simply the best text editor you can buy."

John Dvorak, InfoWorld 7/8/85

Choosing the right program editor is an extremely important decision. After all, more time is spent with your program editor than with any other category of software. A good program editor helps to stimulate creativity and produce programs of superior design. A poor program editor, on the other hand, fights you every step of the way. It can slow you down, get in your way, and make even the simplest tasks difficult.

BRIEF - PROGRAM EDITING YOUR WAY

Every programmer has an individual style that makes their work unique. Most program editors compromise that style by forcing the programmer to conform to their methods. Not so with BRIEF.

BRIEF's most powerful feature is its ability to conform to your way of programming. BRIEF can be easily tailored/customized to your individual preferences. For example, keyboard reassignment allows the keyboard to be used in whatever way you prefer. Keystroke macros allow long sequences of editor commands to be repeated by pressing a single key. Plus you can simultaneously work with numerous program and data files, and configure many windows to control BRIEF's visual presentation.

EXPERT USERS OF EVERY MAJOR EDITOR HAVE SWITCHED!

(Call For Details)

```
make.c
int handle = 0;
main (argc, argv)
int argc;
    fsa.h
#include "..\include\ctype.h"
typedef struct
{
    short action,
    state;
} Fsa;
#ifdef FSA_MAIN
Fsa fsa[10] = { /* Alphabet
/* State 0. */ 0, 2, 10
/* State 1. */ 10, 0, 10
/* State 2. */ 0, 2, 1
/* State 3. */ 0, 5, 11
/* State 4. */ 0, 4, 0
}

makefile.h
/*
**
** This is the definitions fil
** Hopefully, it won't be unreasonab
** that have been written.
**
typedef struct cmd_struct
{
    char *cmd_text;
    struct cmd_struct *next_cmd;
} *Cmd_Ptr, Cmd;
```

Mismatched open parenthesis.

Line: 11 Col: 17 2:17 PM

BRIEF'S ONLY LIMITATIONS ARE THE ONES YOU SET

Try BRIEF with a full 30 day moneyback guarantee.
Only \$195. Call 800-821-2492.

"A BONA FIDE UNDO" Steve McMahon - BYTE REVIEW "BUILD YOUR DREAM EDITOR" MARCH 1985

Here are Steve McMahon's exact words: "...BRIEF implements a true undo facility, by default allowing command-by-command recovery from the last 30 commands...The number of commands you want to be able to undo can be changed." (up to 300) "Only with BRIEF, though, was it possible to undo a macro that produced 4000 words of text with a single keystroke."

EVERY FEATURE YOU WANT

BRIEF supports practically every feature you've ever seen, and some you probably haven't thought of yet.

- Edit Multiple Large Files
- Windows (Tiled & Pop-up)
- Unlimited File Size
- Full UNDO (up to 300)
- True Automatic Indent for C
- Exit to DOS Inside BRIEF
- Compile programs inside BRIEF
- Uses All Available Memory
- Online Help
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning
- Reconfigurable Keyboard
- Adjustable line length (up to 512)
- Search for "Regular Expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- And ... a Complete, Compiled, Programmable and Readable Macro Language.

A TYPICAL BRIEF SCREEN

Notice there are three windows on the screen simultaneously and each one is viewing a different file. The mainline of a C program is visible in the uppermost window; the programmer has run a syntax checking macro which found a mismatched open parenthesis in the arguments to the mainline. The other two windows show header files containing information crucial to the design of the program. BRIEF can have an unlimited number of windows and files accessed simultaneously.

AVAILABILITY: IBM-PC, AT, Compatibles & TANDY 2000

Solution Systems™

335-D Washington St., Norwell, MA 02061 (617) 659-1571 800-821-2492.

*Steve McMahon's quote courtesy Suntype Publishing Systems. BYTE review by Mr. McMahon may be found in BYTE Magazine March 1985.

Adding a COPY Command to ProDOS

by Shawn Day

If you use ProDOS on an Apple or Apple-compatible system, you may have noticed that you cannot copy files from one volume to another without running FILER and losing any BASIC program you have in memory. I ran across this problem when I tried to copy files to the ProDOS RAM disk from within a BASIC program: I wanted my startup program to load all of the ProDOS editor and assembler files into the RAM disk so that I could have the maximum amount of storage space on my single disk drive and speed up the process of assembling long source files. To solve this problem, I wrote a program that adds a COPY command to ProDOS.

the same volume or to copy files from one volume to another. If you are in the process of copying files from one volume to another and no ProDOS prefix has been set, you must specify the complete path name for both files, e.g.,

```
COPY /MYDISK/HELLO,  
      /RAM/STARTUP
```

If a ProDOS prefix has been set, it will automatically be prefixed to either path name if required. For example, if the prefix is /RAM/, the command

```
COPY /MYDISK/HELLO,  
      STARTUP
```

Additional banks of memory can buy you both speed and more usable memory even though your program must reside entirely in one bank.

Using the Program

To add the COPY command to ProDOS, simply type BRUN COPY or use the general-purpose "run" command, -COPY. The code will be installed in memory above the ProDOS file buffers and protected in the system bit map. To execute the command, use the following syntax:

```
COPY pathname1,pathname2
```

A new file (pathname2) will be created, and the contents of the file indicated by pathname1 will be copied into it. You can use this command to make backup copies of any file on

will copy the file /MYDISK/HELLO to the file /RAM/STARTUP. If the destination file name already exists on the destination volume, the command exits with the DUPLICATE FILENAME error. This prevents accidental erasure of files. If you want to copy a file over an already existing file, use the DELETE command to destroy the existing file and then use the COPY command.

COPY cannot be used to copy files directly from one disk to another if you have only one disk drive. However, if you have a ProDOS RAM disk available to you (that is, you are using an Apple IIc or a IIe with an extended 80-column card) you have the capability to copy the file into the RAM disk and then copy the file again from the RAM disk over to the destination disk.

Shawn Day, 724 Glenburn St., Kelowna, British Columbia V1Y 4G6 Canada

Entering the Program

The source code for COPY is shown in Listing One (page 58). To type in the program, you can either use an assembler or type the hexadecimal numbers in from the monitor. When you are finished, save the program with

```
BSAVE COPY,A$6000,L$2B6
```

How it Works

The COPY code is divided logically into three parts: installation, first-stage processing, and second-stage processing. I will describe the operation of each in turn.

Installation

When COPY is invoked, it relocates itself to reside between the ProDOS BASIC.SYSTEM and its file buffers. A call to GETBUFR with the accumulator containing the value \$02 (Listing One, lines 69 and 70) reserves the required two pages of memory. The high byte of the starting address of the reserved space is returned in the accumulator. Line 71 checks to see if the call to GETBUFR was successful. In ProDOS Version 1.0.1 there is a bug in GETBUFR that causes it to return a successful error code whether or not an error occurs. Due to this bug, COPY will only work with ProDOS Version 1.1.1 and later. If you have an earlier version of ProDOS, see your dealer to get an update. If GETBUFR is unsuccessful, the program exits with an appropriate error message (line 72).

Assuming the GETBUFR call was successful, program control is transferred to GOTSPACE (lines 73-83), which protects the acquired memory space via the system bit map and updates RSHIMEM. The routine BITMAPS (lines 135-157) is called twice—once to protect each page. Then the value of RSHIMEM in the BASIC.SYSTEM's global page is decreased by 2 (lines 80-83). This protects the acquired memory from a call to FREBUFR. FREBUFR resets HIMEM to the value contained in RSHIMEM. Because COPY dynamically allocates buffer space during execution via GETBUFR and FREBUFR, it is necessary to perform this

How to go from UNIX to DOS without compromising your standards.

It's easy. Just get an industry standard file access method that works on both.

C-ISAM™ from RDS.

It's been the UNIX™ standard for years (used in more UNIX languages and programs than any other access method), and it's fast becoming the standard for DOS. Why?

Because of the way it works. Its B+ Tree indexing structure offers unlimited indexes. There's also automatic or manual record locking and optional transaction audit trails. Plus index compression to save disk space and cut access times.

How can we be so sure C-ISAM works so well?

We use it ourselves. It's a part of INFORMIX®, INFORMIX-SQL and File-it!™, our best selling database management programs.

For an information packet, call (415) 322-4100. Or write RDS, 4100 Bohannon Drive, Menlo Park, CA 94025.

You'll see why anything less than C-ISAM is just a compromise.



RELATIONAL DATABASE SYSTEMS, INC.

© 1985, Relational Database Systems, Inc. UNIX is a trademark of AT&T. INFORMIX is a registered trademark and RDS, C-ISAM and File-It! are trademarks of Relational Database Systems, Inc.

Circle no. 170 on reader service card.

step. If you wish to have any other added ProDOS commands resident in memory during the execution of COPY, they must each be protected from FREBUFR by this method.

The next program segment (lines 87-95) hooks the COPY command into BASIC.SYSTEM's external command link. Any previously installed external commands are daisy chained by the method described in the *ProDOS Technical Reference Manual*. Lines (99-115) update all internal, nonrelocatable references so that the program will execute properly in its final location. They make use of the table of nonrelocatable references in lines 161-188.

After all the references are adjust-

ed, the code is moved to its final location by the monitor MOVE routine (lines 119-131). This finishes the installation of COPY, and control is passed back to the system.

First-Stage Processing

When ProDOS runs across a command that it does not recognize, it passes control to EXTRNCMD in the BASIC.SYSTEM global page. With COPY installed, this location contains a jump to the relocated copy code (starting at line 203). Here the path-name buffer is checked to see if it contains the COPY command in upper or lower case. If not, the carry flag is set to indicate no match, and control passes to the next user-in-

stalled external command (lines 214-215 and 193). If there are no more user-installed commands, line 193 will contain a jump to an RTS instruction. Assuming the COPY command was found, execution continues at line 218, where the command-string length is decremented and stored in XLEN. Lines 220 and 221 store a 0 in XCNUM to indicate to BASIC.SYSTEM that the command is external, and lines 222-225 store the address where second-stage processing is to be resumed after the input string has been parsed by BASIC.SYSTEM. Next, PBITS and PBITS+1 are set to indicate that file creation is allowed, that the command requires two path names, and that the current prefix should be fetched if none is specified in the command string. Finally, the program exits with the carry flag cleared to tell BASIC.SYSTEM that the command has been identified.

Second-Stage Processing

After BASIC.SYSTEM has parsed the command string, control is passed to line 236. At this point, FBITS and FBITS+1 have been set by BASIC.SYSTEM to indicate which parameters were present in the command string. Lines 236-243 check that two path names were given and exit with a syntax error if not.

Assuming both path names were present, execution continues at PARM_SOK (line 247), where a GET _FILE_INFO call is made to the MLI (machine-language interface) through the GOSYSTEM vector. This call gets the file information for the source file (the first path name specified in the command string). The file type is checked to make sure it is not a directory or bad block file. (Copying these file types would be meaningless.) If the file is either of these two types, the program exits with a FILE TYPE MISMATCH error.

Next, a new file is created with the file name specified by the second path name in the command line (lines 262-277). If an error occurs during the CREATE attempt, control passes to BADCALL (line 277), where the MLI error is translated to a BASIC.SYSTEM error, the carry flag is set,

Oh, Rapture!

This is truly the editor I have been longing for.

- Dr. Joseph Newcomer

New Epsilon 3.0: fast, fully programmable text editor with an EMACS-style command set and concurrent processes!

Presenting Epsilon 3.0, the fastest, most powerful text editor available for personal computers. Epsilon has a built-in programming language, called EEL, for creating your own commands. Plus you get EEL source code for all of Epsilon's commands!

EEL has all the expressive power of the C programming language. It supports all C statements and expressions, pointers and user-defined structures. Unparalleled flexibility!

Because EEL looks like C, commands are easy to write. You don't have to learn a new language. Epsilon detects illegal pointer references, and has a source line single-stepping debugger and EEL profiler, too.

Our amazing Concurrent Process facility lets you run other programs while you continue to edit your files. The program's input and output are connected to a window, so you can edit them. Great for background compiles, debugging while looking at source code, and lots more!

Plus the advanced features programmers need:

- Concurrent Processes
- Multiple Windows
- Unlimited File Size
- Customizable Keyboard
- Tutorial
- Automatic Swap File
- Supports Large Displays
- Saves Deleted Text (n times)
- Context Sensitive Help
- Regular Expression Search
- Unlimited Number of Files
- File Name Completion
- Convenient Keyboard Macros
- Directory Perusal
- Language Support (C, Lisp, etc.)
- Uses All Available Memory

Epsilon runs on IBM PC's, XT's, AT's and compatibles with PC-DOS 2.0 or above and requires 256K of memory.

Epsilon is available directly from Lugaru Software Ltd, and costs only \$195.00. Order now using your Visa, MasterCard, or American Express card. Company purchase orders are also welcome. Order it today, so you can enjoy it soon!

Lugaru Software Ltd.
5740 Darlington Road
Pittsburgh, PA 15217
(412) 421-5911

Circle no. 135 on reader service card.

and an RTS instruction transfers control back to BASIC.SYSTEM. If the CREATE is successful, lines 281-289 attempt to obtain a 1K file buffer from the system for use by the destination file. If the attempt is unsuccessful, the NO BUFFERS AVAILABLE error is generated (lines 284-286).

Finally, the rest of free memory (excluding any resident BASIC programs and any variables that may have values assigned to them) is reserved for the file-transfer buffer (lines 294-316). It is desirable to have as large a buffer as possible in order to speed up the copy. If a very large BASIC program is in memory at the time the COPY command is executed, the file will be copied in several passes because only a small amount of it can be read into memory at one time. If there is no BASIC program in memory, even very large files can be copied in one or two passes. If there is less than one free page of memory, all of the buffer space obtained so far is returned to the sys-

tem, and the program exits with the NO BUFFERS AVAILABLE error (lines 302-305).

Now that all the buffers are set up, lines 320-331 open the destination file, and lines 335-337 open the source file. Any errors detected during this process cause the buffer space to be returned to the system and an appropriate error message to be generated. Lines 346-351 call NEWLINE to indicate that no NEWLINE character is to be recognized. The NEWLINE character is used by ProDOS when reading a file to signify the end of a line or record. For example, when reading lines from a text file, the NEWLINE character should be set to \$8D (carriage return). This allows processing of the file line by line. Because it is desirable to read in as much of the file as possible during each pass, the NEWLINE character is disabled by setting the mask byte to 0 (lines 346 and 347).

Lines 361-378 perform the actual transfer. Lines 363 and 364 call the MLI READ routine to read in as much

of the file as possible. An error during the read causes program control to pass to line 366, where the end-of-file condition is checked. If the end of file has been reached, lines 382-401 close both files, free the allocated buffer space, and return control to the system. Otherwise, lines 369-375 write the buffer contents out to the new file and then (assuming no errors were detected during the WRITE) go back for more (line 376). Lines 402-432 are the parameter lists used during the calls to the MLI.

Improvements and Modifications

The most obvious improvement is to allow the use of disk-to-disk copies with a single disk drive. You might also find it useful to allow COPYING a text file to the screen or a printer. Then you would have a general-purpose copy utility, such as CP/M's PIP command.

DDJ

(Listing begins on next page)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 194.

NEON™ TURN ON THE FULL POWER OF YOUR MACINTOSH™

Hidden within your Mac is the programming power, flexibility, and speed to match your imagination. Neon is your key to this object-oriented world. Based on the same design philosophy as the Mac itself, Neon lets you create and command objects — program modules that you build, perfect, and add to your personal collection of tools. In this Smalltalk-like environment, you wield the power to quickly assemble and test ideas, tuning as you go for maximum speed and efficiency. □ Neon works in 128K or dynamically expands for the fattest Mac possible. It is easy to work with from the very beginning. Additional help is provided through the comprehensive manual by Danny Goodman. □ Created by Kriya Systems, Inc. for the development of our Typing Tutor III™ program, Neon is your answer for professional software development. With Neon, there are no licensing fees. Ever. For the MacIntosh developer, Neon is the best choice.



To order your copy of Neon, call 1-800-34KRIYA now with Visa or MasterCard, or send your check to Kriya Systems, Inc., Six Export Drive, Sterling, VA 22170. The price is \$155, including shipping and handling.

Neon and Typing Tutor III are trademarks of Kriya Systems, Inc. Apple Macintosh is a trademark of Apple Computers, Inc.

Circle no. 100 on reader service card.

TURBO GRAPHICS™

for use with

Borlands Turbo Pascal™

A comprehensive set of Pascal procedures and binary routines that includes most of the graphics capabilities found in BASICA in addition to several new graphics routines.

MAJOR FEATURES

(on medium resolution graphics screen)

- Draw large letters, circles, and lines in color.
- Create complicated graphics using DRAW "macros".
- Create and use WINDOWS non-destructively.
- Paint enclosed areas. Get and Put screen memory.
- Scroll any portion of the screen up or down.
- Use the font editor to create letters and graphics characters.
- Combine features to create interesting animations.

Catalog Numbers:

UTIL-100-IBM PC \$39.95
UTIL-100-IBM PCjr each

Shipping and Handling:

\$5.00 each



Diversified Educational Enterprises, Inc.
725 MAIN STREET
LAFAYETTE, IN 47901

317-742-2690

Circle no. 118 on reader service card.

ProDos Copy Listing (Text begins on page 54)

```

0000: 1 LST GEN,VSVM
0000: 2 *****
0000: 3 *
0000: 4 * COPY Command for ProDOS BASIC.SYSTEM
0000: 5 * by Shawn Day
0000: 6 *
0000: 7 * Syntax: COPY pathname1,pathname2
0000: 8 *
0000: 9 * This command copies the file specified by
0000: 10 * pathname1, to the file specified by pathname2.
0000: 11 * If pathname2 does not exist, it is created.
0000: 12 * If pathname2 does exist, the command exits
0000: 13 * with a DUPLICATE FILENAME error.
0000: 14 *
0000: 15 * Apple ProDOS Assembler version 1.0
0000: 16 *
0000: 17 *****
0000: 003C 19 AIL EQU $3C ;Start address for MOVE
0000: 003D 20 A1H EQU $3D
0000: 003E 21 A2L EQU $3E ;End address for MOVE
0000: 003F 22 A2H EQU $3F
0000: 0042 23 A4L EQU $42 ;Destination for MOVE
0000: 0043 24 A4H EQU $43
0000: 0073 25 HMEM EQU $73 ;HMEM pointer
0000: 00FC 26 TEMP EQU $FC ;Temporary storage
0000: 00FD 27 OFFSET EQU $FD ;Offset for relocation code
0000: 00FE 28 PTR EQU $FE ;Temporary address pointer
0000: 00FE 29 COUNT EQU $FE ;Temporary storage
0000: 0200 30 INBUF EQU $200 ;Keyboard input buffer
0000: BE06 31 EXTRNCMD EQU $BE06 ;External cmd jmp vector
0000: BE09 32 ERRROUT EQU $BE09 ;Print error message
0000: BE50 33 XTRNADDR EQU $BE50 ;External cmd address
0000: BE52 34 XLEN EQU $BE52 ;Length of cmd string-1
0000: BE53 35 XCNUM EQU $BE53 ;cmd no. (extrn. cmd = 0)
0000: BE54 36 PBITS EQU $BE54 ;Command parameter bits
0000: BE56 37 FBITS EQU $BE56 ;Command parameter bits found
0000: BE6C 38 VPATH1 EQU $BE6C ;Pathname 1 buffer pointer
0000: BE6E 39 VPATH2 EQU $BE6E ;Pathname 2 buffer
0000: BE70 40 GOSYSTEM EQU $BE70 ;Perform MLI call
0000: BE8B 41 BADCALL EQU $BE8B ;Translate MLI error to BI error
0000: BE84 42 SGINFO EQU $BE84 ;GET_FILE INFO MLI buffer
0000: BE87 43 FIACCESS EQU $BE87 ;Access used by lock/unlock
0000: BE88 44 FIFILID EQU $BE88 ;File id of disk file
0000: BE89 45 FIAUXID EQU $BE89 ;Aux id
0000: BECE 46 OSYSBUF EQU $BECE ;Buffer for MLI OPEN
0000: BED0 47 ORFNUM EQU $BED0 ;Reference number of opened file
0000: BED2 48 NMLREF EQU $BED2 ;NMLINE file reference number
0000: BED3 49 NMLINL EQU $BED3 ;NMLINE enable mask and char
0000: BED6 50 RWRETRN EQU $BED6 ;Read file reference number
0000: BED7 51 RWDATA EQU $BED7 ;Read buffer
0000: BED9 52 RWCOUNT EQU $BED9 ;Read (Number of char requested)
0000: BEDB 53 RWTRANS EQU $BEDB ;Read (Number of char transferred)
0000: BEDE 54 CPRETRN EQU $BEDE ;Close file reference number
0000: BEF5 55 GETBUFR EQU $BEF5 ;Get buffer from system
0000: BEF8 56 FREEBUFR EQU $BEF8 ;Give buffer back to system
0000: BEFB 57 RSHMEM EQU $BEFB ;HMEM reset value for FREEBUFR
0000: BF00 58 MLI EQU $BF00 ;MLI entry point
0000: BF58 59 BITMAP EQU $BF58 ;System bit map table
0000: FE2C 60 MOVE EQU $FE2C ;Monitor MOVE routine
----- NEXT OBJECT FILE NAME IS COPY
6000: 6000 61 ORG $6000
6000: 62 *
6000: 63 * Installation
6000: 64 *
6000: 65 *
6000: 66 * Get some space for the COPY code
6000: 67 * and protect it by updating the system bit map
6000: 68 *
6000: AD 03 61 69 LDA PAGES ;Number of pages needed
6000: 20 F5 BE 70 JSR GETBUFR ;Ask the BI for some space
6000: 90 03 600B 71 BCC GOTSPACE ;We got the space
6000: 4C 09 BE 72 JMP ERRROUT ;Couldn't find space
6000: 48 73 GOTSPACE 73 PIA ;Save start address
6000: AE 03 61 74 LDA PAGES ;Number of pages to protect
6000: 20 73 60 75 PROTECT JSR BITMAPS ;Protect a page
6000: 18 76
6000: 69 01 77 CLC
6000: CA 78 ADC #01 ;Point to next page
6000: D0 F7 600F 79 DEX ;Are we done?
6000: AD FB BE 80 BNE PROTECT ;If not
6000: 38 81 LDA RSHMEM ;Protect code from FREEBUFR
6000: ED 03 61 82 SEC
6000: 8D FB BE 83 STA PAGES
6000: 22 84 *
6000: 22 85 * Hook routine into the ProDOS external command vector
6000: 22 86 *
6000: AE 07 BE 87 LDX EXTRNCMD+1 ;Set link to next external command
6000: AC 08 BE 88 LDY EXTRNCMD+2 ;or RTS (whichever was already there)
6000: 8E 01 61 89 STX LINK+1
6000: 8C 02 61 90 STY LINK+2
6000: A9 0E 91 LDA #START ;Install the address of the
6000: 8D 07 BE 92 STA EXTRNCMD+1 ;COPY command handler in the
6000: 68 93 PLA ;external command jmp vector
6000: 48 94 PHA ;Save start address for MOVE
6000: 8D 08 BE 95 STA EXTRNCMD+2
6000: 38 96 *
6000: 38 97 * Update all non-relocatable references in the COPY code
6000: 38 98 *
6000: 38 99 SEC
6000: E9 61 100 SBC #LINK ;Find offset to be added to all
6000: 85 FD 101 STA OFFSET ;non-relocatable references
6000: A2 00 102 LDX #00 ;Index start of table
6000: A0 00 103 LDY #00
6000: 18 104 CLC

```

(Continued on page 60)

Another in a series of
productivity notes on
MS-DOS™ software
from UniPress.

Subject: Multi-window full screen editor.

Multiple windows allow several files (or portions of the same file) to be edited simultaneously. Programmable through macros and the built-in compiled MLISP™ extension language.

Features:

- Famed Gosling version.
- Extensible through macros and the built-in compiled MLISP extension language.
- Dozens of source code MLISP functions; including C, Pascal, LISP and MLISP syntax checking.
- EDT and simple WordStar™ emulation modes.
- MS-DOS commands can be executed with output placed in an EMACS window.
- Run a compile on your program and EMACS will point to any errors for ease of debugging.

■ EMACS runs on the IBM-PC™ (XT/AT), TI-PC™, DEC Rainbow 100+, HP-150™ or any other MS-DOS machine. Requires at least 384K.

Price:

EMACS binary \$325
EMACS source 995
One month trial 75
Also available for UNIX™ and VMS™.
Call for pricing.

TEXT EDITING

UNIPRESS EMACS™

ISAM FILE SYSTEM REPORT WRITER & QUERY LANGUAGE

PHACT™

Subject: Powerful keyed file access, report generator and query language for MS-DOS.

PHACT Isam is a keyed B+ tree file manager providing easy access to and manipulation of records in a database. PHACT-report produces formatted reports from PHACT databases. PHACT-query is a high level tool for manipulating data in PHACT databases.

Features:

- Supports fixed and variable length records (1-9999 bytes).
- Up to 9 alternate indices are supported.
- Record locking allows multiple simultaneous updates.
- Records can be accessed on full or partial key.
- Includes full Lattice™ linkable library and high-level functions.
- PHACT-report, an easy-to-use command language for formatting reports from existing PHACT databases.

■ PHACT-query can be used for making ad hoc and "canned" inquiries into a PHACT database.

Price:

PHACT ISAM single-user \$295
multi-user 450
PHACT-report single-user 165
multi-user 275
PHACT-query single-user 195
multi-user 495
Source code available. Call for terms.
Also available for UNIX and VMS.
Call for pricing.

COMMUNICATIONS

Subject: Communication network for PCs.

PCworks connects PCs to a variety of computers and information services.

Features:

- PCworks connects your PC to other PCs/Macintoshes™ or UNIX/VMS systems for file transfer and command serving.
- Menu driven for ease of use. Telephone numbers and other parameters can be stored.
- Transfer ASCII or binary files, send/receive electronic mail, or submit and print jobs in conjunction with UNIX or VAX™/VMS hosts.
- PC files can be printed on another system.
- Terminal Emulators include: ANSI, TTY, VT100/52.
- Protocol: error-free data transfers utilizing packet sequence numbers and checksums, or XMODEM protocol.
- Supports many modems/autodialers at a wide range of speeds.

Price:

PCworks \$195
The Connectable Networks™
Call for pricing.

For our **Free Catalogue** and more information on these and other software products, call or write:
UniPress Software, Inc.;
2025 Lincoln Hwy., Edison, NJ 08817.
Telephone: (201) 985-8000.
Order Desk: (800) 222-0550
(Outside NJ). Telex 709418.
European Distributor: Modulator SA,
Switzerland Telephone: 41 31 59 22 22,
Telex: 911859

OEM terms available.
Mastercard/Visa accepted.

See our ad in the next issue for more MS-DOS products, including Lattice C compiler and Ps-Make.

PCworks™

New from UniPress!

Trademarks of UniPress: EMACS and MLISP. UniPress Software, Inc., WordStar, MicroPro Int'l. Corp., MS-DOS and IBM PC, IBM, TI-PC, Texas Instruments, PHACT, PHACT Associates, Lattice, Lattice, Inc., UNIX, AT&T Bell Laboratories, VAX/VMS, DEC Rainbow 100+, Digital Equipment Corp. HP-150, Hewlett-Packard Co. MacLine, UniHost, PCworks and The Connectable Network, Touchstone Software, Corp. Macintosh, Apple Computer, Inc.

UniPress Software

C POWER

FUNCTION LIBRARY C WINDOWS SUPERFONTS FOR C B-TREE LIBRARY ISAM DRIVER C-TERP INTERPRETER

LIBRARY: 325 fully tested functions
screen handling/graphic, cursor/
keyboard/data entry, string handling,
status and control, utility/DOS/BIOS/
time/date, printer control.
More & Better Functions!
4 disks, 400 page manual \$129.95

WINDOWS: The most powerful
windowing package available,
overlays, borders, pop-up menus/
help windows, zap instantly on/off
screen, status lines, horizontal/
vertical scrolling, color control/
highlighting, word-wrap, files to
windows, keyboard to windows
3 disks, 225 page manual \$129.95

SUPERFONTS FOR C: Dramatic,
high impact screens. Use our
character & image libraries and
functions, or create your own.
Font & Function Library \$49.95

B-TREE LIBRARY: Fast index-
ing B-tree. 16 million each: keys and
records, unlimited keys per file,
variable length records, configurable
key sizes. PC-DOS, UNIX, XENIX,
CP/M. \$79.95

ISAM DRIVER: \$49.95

COMPATIBILITY: PC/XT/AT &
compatibles; Full K&R, CI-C86/De
Smet/ Lattice/Microsoft 3.0/Aztec/
others. BEST YOU CAN GET!

VALUE: All source code, No
royalties, on packages above.

C-TERP: Simply the best C
interpreter available. See C Journal
Summer 1985

C-TERP (specify compiler) \$299.95

COMBINE AND SAVE!

C LIBRARY plus C WINDOWS

BOTH for only \$179.95

+ SUPERFONTS FOR C \$199.95

+ B-TREE and ISAM \$299.95

(A \$440 VALUE)

C LIBRARY plus C WINDOWS

+ SUPERFONTS

+ C-TERP (a \$810 value) \$459.95

All (a \$740 value) \$549.95

Entelekon

12118 Kimberley, Houston, TX 77024
713-468-4412

VISA-MASTECARD-CHECK

ProDos Copy Listing

(Listing continued, text begins on page 54)

```

6042:BD 91 60 105 RELOCATE LDA RELTABLE,X ;Point to non-relocatable reference
6045:05 FE 106 STA PTR
6047:BD 92 60 107 LDA RELTABLE+1,X
604A:85 FF 108 STA PTR+1
604C:B1 FE 109 LDA (PTR),Y ;Add offset to the reference
604E:65 FD 110 ADC OFFSET
6050:91 FE 111 STA (PTR),Y
6052:E8 112 INX ;Index next reference
6053:E8 113 INX
6054:0C C7 60 114 CPX TABLESIZE ;Number of references times 2
6057:90 E9 6042 115 BCC RELOCATE
6059: 116 *
6059: 117 * Move the code to its final home
6059: 118 *
6059:AZ 00 119 LDX #>LINK ;Start
605B:A0 61 120 LDY #<LINK
605D:86 3C 121 STX ALL
605F:84 3D 122 STY All
6061:A2 AB 123 LDX #>END ;End
6063:A0 62 124 LDY #<END
6065:86 3E 125 STX A2L
6067:84 3F 126 STY A2H
6069:A0 00 127 LDY #00 ;Destination (Always on a page boundary)
606B:84 42 128 STY A4L
606D:68 129 PLA ;We saved the high byte on the stack
606E:85 43 130 STA A4H
6070:4C 2C FE 131 JNP MOVE ;Y-reg=$00
6073: 132 *
6073: 133 * Subroutine to update system bit map
6073: 134 *
6073:AB 135 BITMAPS TAY ;Save page address
6074:48 136 PIA
6075:8A 137 TXA ;Save X-reg
6076:48 138 PIA
6077:90 139 TYA ;Restore page address
6078:4A 140 LSR A ;Divide page address by #
6079:4A 141 LSR A
607A:4A 142 LSR A
607B:AA 143 TAX ;Load X-reg with page address/8
607C:98 144 TYA ;Restore page address
607D:29 07 145 AND #07 ;Kill off the high order bits
607E:A8 146 TAY ;and put it in the Y-reg
6080:A9 00 147 LDA #00 ;Zero the mask
6082:38 148 SEC ;Prepare to put a 1 into mask
6083:6A 149 BITMAPS1 ROR A ;Rotate carry into mask
6084:83 150 DEY
6085:10 FC 6083 151 BPL BITMAPS1 ;Rotate mask to proper position
6087:1D 58 BF 152 ORA BITMAP,X ;Update the actual bitmap
608A:9D 58 BF 153 STA BITMAP,X
608D:68 154 PLA ;Restore X-reg and A-reg
608E:AA 155 TAX
608F:68 156 PLA
6090:60 157 RTS
6091: 158 *
6091: 159 * Table of non-relocatable references
6091: 160 *
6091:1F 61 161 RELTABLE DW RELOC1+2
6093:24 61 162 DW RELOC2+2
6095:2C 61 163 DW NXTCHR2+2
6097:3E 61 164 DW RELOC3+1
6099:78 61 165 DW FILEBK+2
609B:7E 61 166 DW RELOC4+2
609D:84 61 167 DW RELOC5+2
609F:89 61 168 DW RELOC6+2
60A1:8F 61 169 DW RELOC7+2
60A3:95 61 170 DW RELOC8+2
60A5:9B 61 171 DW RELOC9+1
60A7:AE 61 172 DW RELOC10+2
60A9:B3 61 173 DW RELOC11+2
60AB:D6 61 174 DW RELOC12+2
60AD:DF 61 175 DW RELOC13+2
60AF:E7 61 176 DW RELOC14+2
60B1:ED 61 177 DW RELOC15+2
60B3:F3 61 178 DW RELOC16+2
60B5:F9 61 179 DW RELOC17+1
60B7:3B 62 180 DW TRANSFER+2
60B9:3E 62 181 DW RELOC18+2
60BB:51 62 182 DW RELOC19+2
60BD:57 62 183 DW RELOC20+2
60BF:5D 62 184 DW RELOC21+1
60C1:74 62 185 DW RELOC22+2
60C3:77 62 186 DW RELOC23+2
60C5:7D 62 187 DW RELOC24+1
60C7:36 188 TABLESIZE DFB TABLESIZE-RELTABLE
60C8: 0038 189 FILLPAGE DS >0-FILLPAGE ;Fill to next page boundary
6100: 190 *
6100: 191 * The following is the COPY command code.
6100: 192 *
6100:4C 00 00 193 LINK JMP $0000 ;Address planted to next command
6103:02 194 PAGES DFB <END-LINK+256 ;Number of pages occupied
6104:04 43 4F 50 195 CHDSTRU STR "COPY" ;This is the command string
6108:59
6109:04 63 6F 70 196 CHDSTRL STR "copy" ;Accept upper or lower case
610D:79
610E: 197 *
610E: 198 * Check to see if the command is COPY. If not,
610E: 199 * then check for next user command, or tell PRODOS
610E: 200 * it is not a user command. If it is the COPY
610E: 201 * command, then set up for first stage processing.
610E: 202 *
610E:AD 6C BE 203 START LDA VPATH1 ;Get command line pointer
6111:85 FE 204 STA PTR
6113:AD 6D BE 205 LDA VPATH1+1
6116:85 FF 206 STA PTR+1

```

(Continued on page 62)

Circle no. 92 on reader service card.

Prospero's FIPS-validated
Pascal language compiler
system

WE SPECIALISE
IN WRITING
COMPILERS

8087

Optional 8087 support included

NO BUGS
YOU DON'T
HAVE TO
FILL IN ALL
THE HOLES

"Superb
Quality"

7^{digit}
16

Single and double
precision IEEE
format arithmetic

Pro Pascal allows chaining and
execution of other programs.



Produces compact
efficient code

U.S.A.

Distributed by Lifeboat Ass NY
and Programmer's Shop MA

Lazy Overlaying

Unique fluid overlaying technique
loads non-resident routines
automatically

Z80
family

8086
family

68000
family

Portable source code

PRO PASCAL *works*

Small or
Large

Choice of small or
large model 8086 code

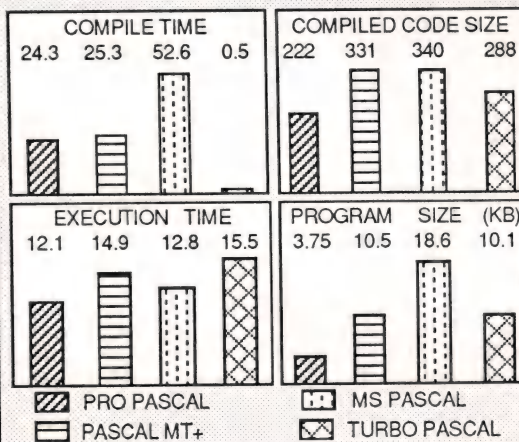
Used at 3000 sites
around the world

Lazy I/O

All files closed on
exit from procedure

Validated to ANSI/IEEE 770X3.97 Class A
on the IBM Personal Computer - the only
validated Pascal compiler available for the
PC and other 8086 family machines!

BYTE "SIEVE" BENCHMARK, IBM PC



Compiles BIG programs
>10000 lines... >1000 identifiers....
Separate compilation to build libraries
and massive program suites

\$390.00

To buy call London
01-741 8531. Visa,
Amex, Mastercard &
Diners accepted.

Powerful symbolic
debugger included

Display and alter variables. Show
source as program runs, or on
request. Break on variables or
procedure calls. Show calling nest.

Prospero Software

Languages For Microcomputer Professionals

190 Castelnau, London SW13 9DH, England telephone 01-741 8531 telex 8814396

EARTH LAUNCHES NEW STARS

EARTH COMPUTERS launches two of the "Hottest" new stars in the S-100 Universe. Both the **TURBOMASTER 8™** and **TURBOSLAVE I™** are Star performers, featuring high speed Z-80H CPUs.

EARTH's new stars are fully compatible with the Multi-user TurboDOS operating system, and will operate in most S-100 systems, including pre-IEEE 696 systems.

TURBOMASTER 8

This outstanding new 8-Bit Single Board Computer offers features that are out of this world:

- On-board ST-506 Winchester Controller
- TurboDOS, CP/M, MP/M compatible
- 5-1/4" and 8" Floppy Controller
- Up to 256KB of memory • 2 Serial ports
- 8 MHz, Z-80H CPU • 1 Parallel port

TURBOSLAVE I



The perfect companion to the TURBOMASTER 8 or other 8/16-Bit Master processor. This high speed slave utilizes an 8 MHz Z-80H CPU and offers extensive on-board diagnostics... an industry exclusive.

- No paddle boards
- S-100, IEEE 696 compatible
- 128KB of RAM
- 2 RS-232 ports, 50-38.4K Baud
- FIFO communications

EARTH COMPUTERS also manufactures a growing line of PC compatible stellar performers such as:

- **TURBOSLAVE PC™**—an 8 MHz



Z-80 single board slave processor that runs CP/M applications on a PC and is compatible with the TurboDOS multi-user operating system.

- **TURBOACCEL 286™**—a high performance 80286 accelerator that boosts PC performance up to five times.

- **EARTHNET PC™** and **EARTHNET S-100**, the low cost, ARCNET-compatible way to tie PC and S-100 systems together.

To put these stars to work for you, call or write EARTH COMPUTERS. **BE SURE TO ASK ABOUT HOW YOU CAN WIN A FREE Z-80 CO-PROCESSOR BOARD.**

EARTH COMPUTERS

"Building Blocks For The Super Micro"

P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV
PHONE: (714) 964-5784

Circle no. 179 on reader service card.

ProDos Copy Listing

(Listing continued, text begins on page 54)

```

6118:A0 00      207      LDY    #$00      ;Index command string
611A:C8         208      INY     ;Increment Index
611B:B1 FE     209      LDA     (PTR),Y ;Get character from input string
611D:D9 04 61  210      CMP     CMDSTRU,Y ;Does it match upper case?
6120:P0 08 612A 211      BPC     NCTCHR2 ;If yes
6122:D9 09 61  212      CMP     CMDSTRU,Y ;Does it match lower case?
6125:P0 03 612A 213      BPC     NCTCHR2 ;If yes
6127:38        214      SEC     ;Flag match failure
6128:D6 06 6100 215      BCS     LINK   ;Go to next command in the chain
612A:CC 04 61  216      CPY     CMDSTRU ;Checked all characters?
612D:00 ED 611A 217      LNE     NCTCHR ;No, so check the next one
612F:88        218      DEY     ;Put command length-1 in XLEN
6130:8C 52 BE  219      STY     XLEN
6133:A9 00     220      LDA     #$00      ;Mark command as external
6135:8D 53 BE  221      STA     XCNUM
6138:A9 4E     222      LDA     @SECOND ;Point to command handler
613A:8D 50 BE  223      STA     XTRNADDR
613D:A9 61     224      LDA     @SECOND
613F:8D 51 BE  225      STA     XTRNADDR+1
6142:A9 08     226      LDA     $0B      ;Allow CREATE, require two pathnames
6144:8D 54 BE  227      STA     PBITS
6147:A9 04     228      LDA     $04      ;Fetch prefix, if required
6149:8D 55 BE  229      STA     PBITS+1
614C:18        230      CLC     ;No errors so far
614D:60        231      RTS     ;Return to BASIC.SYSTEM
614E:          232      *-----
614E:          233      * COPY second stage processing. First check PBITS
614E:          234      * to see if the proper parameters were present.
614E:          235      *-----
614E:AD 56 BE  236      SECOND LDA     PBITS ;Parameters found
6151:4A        237      LSR     A ;Filename found?
6152:90 03 6157 238      BCC     SYNERROR ;No, so give error message
6154:4A        239      LSR     A ;Second filename found?
6155:B0 04 615B 240      BCS     PARMOK ;Yes, so go do command
6157:A9 10     241      SYNERHOR LDA     $10 ;Syntax error
6159:38        242      MLIERR  SEC     ;Set carry to show error
615A:60        243      RETURN  RTS     ;Return to system
615B:          244      *
615B:          245      * Both filenames were given, so execute COPY
615B:          246      *
615B:A9 0A     247      PARMOK  LDA     $0A ;Set up GET_FILE_INFO call
615D:8D B4 BE  248      STA     SSGINFO
6160:A9 C4     249      LDA     $C4 ;MLI GET_FILE_INFO code
6162:20 70 BE  250      JSR     COSYSTEM ;Perform the MLI call
6165:B0 F3 615A 251      BCS     RETURN ;If error, return carry set
6167:AD B8 BE  252      LDA     FIFILID ;Get file type
616A:C9 0F     253      CMP     $0F ;Can't copy a directory file
616C:F0 04 6172 254      BEQ     MISMATCH ;Can't copy a bad block file
616E:C9 01     255      CMP     $01
6170:D0 04 6176 256      BNE     FILEOK ;FILE TYPE MISMATCH error
6172:A9 0D     257      MISMATCH LDA     $13 ;Always
6174:D0 E3 6159 258      BNE     MLIERR
6176:          259      *
6176:          260      * File type is OK
6176:          261      *
6176:8D 93 62   262      FILEOK  STA     CNFILID ;File type for CREATE
6179:AD B9 BE  263      LDA     FIAUXID ;File aux type for CREATE
617C:8D 94 62   264      RELOC4  STA     CRAUXID
617F:AD BA BE  265      LDA     FIAUXID+1
6182:8D 95 62   266      RELOC5  STA     CRAUXID+1
6185:A9 C3     267      LDA     $C3 ;File access (UNLOCKED)
6187:8D 92 62   268      RELOC6  STA     CRACCESS ;We have to be able to write!
618A:AD E6 BE  269      LDA     VPATH2 ;Pathname for CREATE
618D:8D 90 62   270      RELOC7  STA     CRPATH
6190:AD F6 BE  271      LDA     VPATH2+1
6193:8D 91 62   272      RELOC8  STA     CRPATH+1
6196:20 00 BF   273      CREATE  JSR     MLI ;Create the new file
6199:CV        274      DFB     $C0 ;MLI CREATE command code
619A:8F 62     275      RELOC9  DW     CRPARMS ;CREATE parameter list
619C:90 03 61A1 276      BCC     SUCCESS ;CREATE successful
619E:4C 8B BE  277      JMP     BADCALL ;Translate to BI error code and exit
61A1:          278      *
61A1:          279      * Create a file buffer for the destination file
61A1:          280      *
61A1:A9 04     281      SUCCESS  LDA     $04 ;1K buffer required
61A3:20 F5 BE  282      JSR     GETBUFR ;Get space
61A6:90 04 61AC 283      BCC     RELOC10
61A8:A9 0C     284      LDA     $0C ;NO BUFFERS AVAILABLE error
61AA:38        285      SEC     ;
61AB:60        286      RTS     ;
61AC:8D 9F 62   287      RELOC10 STA     OPEN2BUF+1
61AF:A9 00     288      LDA     $00 ;Buffer always on page boundary
61B1:8D 9E 62   289      RELOC11 STA     OPEN2BUF
61B4:          290      *
61B4:          291      * Create a transfer buffer for the file
61B4:          292      * Use as much free memory as possible
61B4:          293      *
61B4:A9 FF     294      TRANSBUF LDA     $FF ;Count how many pages we can get
61B6:85 FE     295      STA     COUNT ;Initialize with -1
61B8:E6 FE     296      GETMORE  INC     COUNT ;Count how many pages
61BA:A9 01     297      LDA     $01 ;Get one page at a time
61BC:20 F5 BE  298      JSR     GETBUFR ;Get space
61BF:90 F7 61B8 299      BCC     GETMORE ;Keep going until we can't get more
61C1:A5 FE     300      LDA     COUNT ;Check how many pages we got
61C3:D0 07 61CC 301      BNE     GOTENOUGH ;Any non-zero number is okay
61C5:20 F8 BE  302      JSR     FREEBUFR ;Give back the allocated memory
61C8:A9 0C     303      LDA     $0C ;NO BUFFERS AVAILABLE error
61CA:38        304      SEC     ;
61CB:60        305      RTS     ;
61CC:8D DA BE  306      GOTENOUGH STA     BWCOUNT+1 ;Save buffer size
61CF:A9 00     307      LDA     $00 ;Length is always an even number of page
61D1:8D D9 BE  308      STA     BWCOUNT
61D4:AD 9F 62   309      RELOC12 LDA     OPEN2BUF+1 ;Get start address of buffer
61D7:38        310      SEC     ;

```

(Continued on page 66)

NEW !!

TURBO

NEW !!

INTEGRATED DEVELOPMENT SYSTEM

These systems were developed from the ground up to work in an integrated fashion ... This saves you time and money from trying to assemble different packages and to code in our Products Unique Features.

TURBO ISAM MASTER

GENERATES 'Ready to Run' Turbo Programs Using BORLAND'S Turbo Pascals Database TOOLBOX.

With a few key strokes you can generate the following Pascal programs/Includes files. Automatically interfaces to Turbo Screen Master.

A. MASTER DATABASE PROGRAM - Generates Pascal Program Code for the following functions.

1. **Add a Record** - Allows both duplicate & unique only keys. • Change the up to 6 fields comprising the primary key and the up to 3 secondary keys at any time during input, and the validity of the keys are checked and the ISAM key files are automatically adjusted.

2. **Delete a Record** - Shows the record to be deleted on the screen and allows you to change your mind about deleting, and then adjusts the keys files automatically for the up to 4 keys. The disk space is reused automatically by the programs.

3. **Edit a Record** - with a key change allowed.

4. **Search Database by Key** - Up to 4 keys-and has a 1 key lookup to find next key in alphabetic order.

5. **For Each Key** - The ability to display the keys on the screen or printer in sorted order.

B. DATA BASE RECOVERY PROGRAM. This program recovers your data base if it is corrupted by a power outage or certain hardware failures.

C. MULTIPLE ISAM FILES - Can be used at the same time in a single program. The generator also produces context sensitive instructions on how to integrate the generated program into your main program.

D. DOCUMENTATION - Print screens and ISAM specifications. Also inline program documentation is generated.

TURBO MENU MASTER

GENERATES 'Ready to Run' Turbo Programs with Dedicated Screens or Windows.

Features: Has Menu Data Base so its easy to modify menus.

Provides Selection by - (1) Press of a number-if you have more than 9 selections then you enter only a single number if the selection is 2-9. (2) Press of a function key - if more than 10 selections then can enter use shift function key to get up to 18 selections. (3) Press the High Lighted Letter -In the selection description, you choose a letter for each menu selection in the interactive menu generator. (4) Use the arrow keys - to point to a menu item & then press return.

Adaptive Screen Colors -Different screen colors are automatically used for color or monochrome display. This allows you to provide a better interface for the user.

Interactive Menu Builder - (1) Allows for the automatic entry & reorder of selections. (2) Offers easy color selection. (3) Allows for choice of procedure, chain, execute or comment code generation for each selection. (4) Has startup Menu that YOU design. (5) Provides for the integrated Display & Control of the Key Lock Status.

RISK FREE TRIAL

Try the demo package Included for 30 days. If not pleased return for a full refund.

Credit card orders call:

1-800-821-9503

In Florida (800) 342-0137

NO ROYALTIES
on Generated
Programs

HAWAIIAN VILLAGE COMPUTER SOFTWARE
1109 Pennsylvania Ave. St. Cloud, Florida 32769

Circle no. 133 on reader service card.

TURBO SCREEN MASTER

Some of the **UNIQUE** features of the Turbo Screen Generator are:
Intelligent Cursor - Computer automatically determines which single/double character to use from the Extended IBM Character set to draw boxes and borders. You just move the cursor where you want the border. The other Screen Systems require that you must tediously choose the correct graphic character.

Extended Variable Names - Allows the use of Structures & Arrays in your variable names and automatically generates the correct declarations and initialization routines. Also you are allowed to declare a field larger than the screen display size.

Provides Three Color Selections Methods - Easy to develop color screen on Monochrome Monitor and to develop Monochrome Screens on the Color Monitor.

Adaptive Screen - Entry colors change automatically depending on what type of screen is in use.

COMPARISON WITH OTHER PRODUCTS	TURBO SCREEN MASTER	TURBO SCREEN VER - 1.10	SCREEN SCULPTURE Ver - 1.01
Border Color Control	YES	NO	NO
You Assign Variable Names	YES	NO	YES
No cyptic variable names whose names depend on where they appear on screen			
Range and Date Checks	YES	NO	YES
Data Entry Valid Character Set	YES	NO	NO
Data Entry Mask	YES	NO	YES
Helpful for Profession Screen Input & Validation			
Initialize Variables to a starting value	YES	NO	YES
Data Entry Valid String Set	YES	NO	NO
Pascal storage for type of Boolean & Integer	YES	YES	NO
Control Caps/Num Lock	YES	NO	NO
Auto-Initialization of Date/Time	YES	NO	NO
User Defined Error & Message Handler	YES	NO	NO
Generated program adapts automatically to IBM Screen-Monitor Type	YES	NO	YES
Handles Function Keys	YES	NO	NO
Help Screen Procedures	YES	YES	NO
Optional ISAM Keys Screens Code Generated automatically	YES	NO	NO
Turbo Toolkit Included	YES	NO	NO
Undo Function	YES	NO	NO
Provides Running Time Display	YES	NO	NO
Integrated Data Dictionary	YES	NO	NO
Auto-start Screen for Screen Generation	YES	NO	NO
PRICE (INCLUDES SHIPPING)	\$89.95	\$4.95	125.00

FREE TURBO TOOL KIT \$44.95

VALUE

WITH ORDER OF A MASTER PRODUCT

YES, Enclosed is

INTRODUCTORY OFFER

<input type="checkbox"/> Turbo Pascal 3.0	\$69.95	\$59.95
<input type="checkbox"/> Turbo DatabaseTool Box	\$54.95	\$49.95
<input type="checkbox"/> Turbo Menu Master	\$69.95	\$54.95
<input type="checkbox"/> Turbo Screen Master	\$124.95	\$89.95
<input type="checkbox"/> Turbo ISAM Master	\$149.95	\$124.95

Systems Requires: IBM PC/XT/AT or 100% Compatible -198K, MS DOS 2.0 or Higher - 80 column screen, Turbo Pascal 3.0 - 2 DD/D5 Disk Drives. ISAM Program Generator Requires Turbo Database Toolbox.

*These prices include shipping to all U.S. Cities. All foreign orders add \$10 per product ordered

For Other Inquires Call (305) 892-5686

Name: _____ Phone: _____

Shipping Address: _____

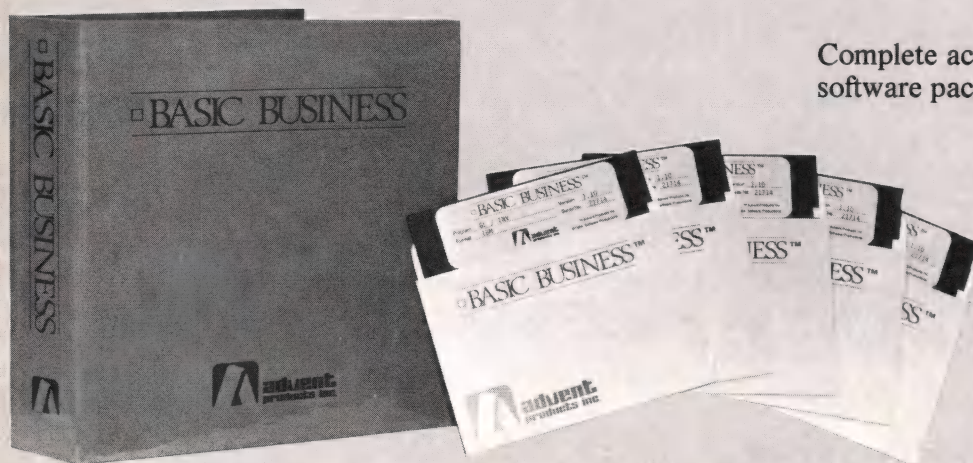
City: _____ State: _____ Zip: _____

VISA or MC #: _____ Exp. Date: _____

C.O.D.'s and Purchase orders WILL NOT be accepted. Outside USA: make payment by bank draft payable in US dollars drawn on a US bank.

©Turbo Pascal & Turbo Database Toolbox are trademarks of Borland International. IBM is a trademark of International Business Machines. MS-DOS is a trademark of Microsoft. Turbo Screen is a copywrite of Pascom Computing. Screen Sculpture is a trademark of the Software Bottling Co of New York. ©1985 Hawaiian Village Computer Software.

TIMELY REPORTS, ACCURATE RECORDS, RELIABLE INFORMATION.....



Complete accounting
software package, only

\$ 89⁹⁵

.....this is ☐ **BASIC BUSINESS**

Price is nice - and so is a long list of features - but when you choose a software package to automate your accounting, don't lose sight of the basics of good business record keeping. Basic Business, an all-in-one accounting software package consisting of General Ledger, Accounts Receivable, Accounts Payable, Payroll, Inventory Control, Sales Order Processing, and Purchase Order Processing. It offers an excellent price and extensive feature list without sacrificing these basic business principles:

Full Audit Trails - to give you complete confidence in the accuracy of your data and provide crucial backup information when you need it. And double entry accounting keeps your books in balance.

Complete Integration - provides efficient processing for all your business transactions, updating all of your accounting records instantaneously. You only enter the transaction once, and all supporting modules are updated automatically.

Flexibility - Basic Business can be adapted to your way of doing business, including balance forward or open item accounts receivable, departmental or consolidated general ledger, variable aging periods and easy entry of manually written checks and voids.

Extensive Reporting - each accounting module provides complete reports, including master lists, transactions, journals, statements and forms. In addition, all data files are compatible with Ashton-Tate's dBase II and dBase III, for the ultimate in custom reporting capability.

With some accounting software, even packages costing hundreds of dollars more, you must make detailed estimates and complex calculations for the maximum number of customers, transactions, inventory items, etc., before using the system. Then, when your business grows to exceed these original estimates, you must repeat this process.

Not with Basic Business! All data files are automatically initialized when you install the system on your computer. Files can grow dynamically as your business increases and are usually limited only by the amount of you exceed your original estimates.

FORMS ARE NO PROBLEM! Basic Business uses standard forms for invoices, statements, checks, purchase orders, etc., which may be ordered with your company name, address and logo imprinted.

Basic Business is one of a family of accounting and business software packages, and has sold previously for several hundred dollars per module. It has been improved, updated and re-packaged to sell at a market-busting \$89.95 for all seven modules. A Point-of-Purchase module, which controls an electronic cash drawer and allows direct entry of transactions from your sales counter is available. Also a dBase file format program is available for importing Basic Business data files into dBase II/III for custom report generation or other special uses.

SOURCE CODE AVAILABLE

Do you have a distinct accounting problem that off-the-shelf software won't handle? Special forms or statements? Don't write your own accounting system from the ground up - start with Basic Business. Call (714) 630-0446 for all the details on source code licensing.

Basic Business can go to work for you today and is available for most popular MS-DOS (IBM and compatibles) and CP/M-80 personal computers. Compare our price, features and attention to detail. There is only one choice. ... it's Basic Business.

Basic Business	\$89.95
Point-of-Purchase module	\$99.95
dBase II/III file formats	\$19.95

Minimum hardware Required for MS-DOS: 128K memory, two 360K floppy drives (hard disk recommended for Sales and Purchase Order Processing), 132 column printer, MS-DOS (or PC-DOS) version 2.0 or later.

Minimum Hardware Required for CP/M-80 computers: 80 x 24 character display terminal, 64K memory, two 360K disk drives (hard disk recommended for Sales and Purchase Order Processing), 132 column printer.

SERVICES

File Transfer Service: Advent provides a service beyond the ability of any format conversion software! We can transfer files between MS-DOS/PC-DOS, CP/M and other operating systems in 300 different 3 1/2", 5 1/4" and 8" formats. Includes Apple and Mac, Apricot, Data General One, Kaypro 2000, Eagle, Epson QX-10 & PX-8 (ROM), HP-150, and North Star computers.

ENGINEERING SOFTWARE

ACNAP: A stand-alone Electronics Circuit Analysis Program for use with passive and active circuits consisting of resistors, capacitors, inductors, transistors, op-amps, FETs, etc. Features menu driven and very fast processing times with circuits saved to disk for later use or editing.

ACNAP (CP/M & MS-DOS) \$69.95

DCNAP: Stand-alone DC circuit analysis program for use with passive and active circuits containing resistors, voltage sources, independent and dependent current sources. Fast, menu-driven program with circuit saved to disk for later use or editing.

DCNAP (CP/M & MS-DOS) \$69.95

Plotpro: Scientific graph printing program. Prints on 80 or 132 column printer. Create linear, semi-logarithmic, and full logarithmic plots with one or two Y axes in auto or forced scale.

Plotpro (CP/M & MS-DOS) \$69.95

SPP: This Signal Processing Program contains an integrated set of routines which analyze linear and non-linear systems and circuits and their effects on user specified time domain waveforms. Based on a 512 point Fast Fourier Transform and its inverse. Signal processing is in frequency domain and non-linear processing is in time domain.

SPP (CP/M & MS-DOS) \$69.95

SOFTWARE UTILITIES

Autodiff: File difference detector. This program finds insertions, deletions, and changes between any two files. Autodiff can mark the file, display, or print the differences, and more!

Autodiff (CP/M) \$29.95

CP/M DateStamper: Automatically stamp your files with the date it is created, last read, or modified. Works without a Real Time Clock, or with many clocks currently on the market. Utilities are included to allow copying, erasing, or renaming files based on time and date. A time logging utility is included to record computer usage for business/tax purposes.

DateStamper (CP/M) \$49.95

Media Master +: Read and Write up to 75 CP/M, MS-DOS & TRSDOS disk formats on your IBM or look-alike computer. ZP/EM program is bundled with Media Master to allow CP/M programs to run directly on your MS-DOS computer. An \$80.00 value.

Media Master + \$59.95

Pack and Crypt: Two program set. Pack compresses and expands files on disk to save space. Crypt encodes files to provide security for sensitive data. Both are ideal for use with modem transfers.

Pack and Crypt (CP/M & MS-DOS) \$24.95

Sidekick: One of the most popular programs ever written. Use Sidekick as a calculator, notepad, appointment calendar, auto dialer, ASCII conversion table and much more. On-line help if you forget any of Sidekick's many functions.

Sidekick (MS-DOS) \$54.95

SmartKey II: New Release! Same great time saver as the original, and allows compiling of definitions you set up with your word-processor! Makes every software program you use easier. Can reduce keystrokes by more than 50% by redefining any key on your keyboard to be any combination of characters or commands that you desire.

SmartKey II \$49.95

SmartPrint: A powerful add-on to SmartKey, SmartPrint is a versatile writing tool designed to give you full access to your printer's features such as wide, bold, condensed, underlined, subscript, superscript, and more. Works great with programs like WordStar and others.

SmartPrint \$29.95

Uniform: Your Computer can read and write up to 80 CP/M, MS-DOS / PC-DOS & TRSDOS disk formats. Versions available for most popular CP/M and MS-DOS computers. Specify your host computer when ordering.

Uniform (CP/M & MS-DOS) \$69.95

XTREE: Directory maintenance program that graphically displays subdirectories and filename paths. Complete control of your directory including delete, rename, view, list or show. A must for your IBM or compatible.

XTREE (MS-DOS) \$49.95

Super Zap: Disk patch and dump program. If you have used DU, you will love this menu driven marvel!

Super Zap (CP/M) \$24.95

ZP/EM: Run almost any CP/M program on your IBM or clone. Use with Media Master or Uniform to allow programs on CP/M disk formats to run directly on your IBM or compatible computer.

ZP/EM (MS-DOS) \$39.95

FX, QX-10, PX-8 - Epson Corp.; CP/M - DRI; MS-DOS - MicroSoft; PC-DOS - IBM Corp.; dBASE II & dBase III - Ashton-Tate; WordStar - MicroPro; UNIX - Bell Laboratories; Apple - Apple Computer Inc.; Basic Business - Advent Products Inc.

PROGRAMMING LANGUAGES

C/80 Ver 3.1: Full featured C compiler and runtime library. One of the fastest on the market. Mathpak is included for true 32 bit floating point and signed integers.

C/80 Ver. 3.1 (CP/M) \$79.90

C/NIX: Operating System Enhancement for CP/M. C/NIX gives your system many features in the UNIX OS such as hierarchical directory, I/O redirection, "pipes" & "filters" and command files. Uses only 2.3K of TPA and 42K of disk. Requires CP/M 2.x.

C/NIX \$59.95

LISP/80: Experiment with the artificial intelligence language. Based on the INTERLISP dialect, LISP/80 offers over 75 built-in functions, including file I/O, and string operations. Complete with 36 page manual and demo programs.

LISP/80 \$39.95

Toolworks C: This compiler is a complete subset of C. The two-pass compiler produces relocatable object files (.obj) which are compatible with the MS-DOS LINK program. Mathpak is included for true 32 bit floating point and signed integers.

Toolworks C Compiler (MS-DOS) \$79.90

Turbo Pascal: Borland version 3.0. The best Pascal compiler on the market.

Turbo Pascal (CP/M & MS-DOS) \$69.95

Turbo Toolbox: Set of 3 utilities for use with Turbo Pascal.

Turbo Toolbox (CP/M & MS-DOS) \$54.95

Turbo Tutor: Teaches step-by-step how to use Turbo Pascal.

Turbo Tutor (CP/M & MS-DOS) \$34.95

Turbo Graphics: Provides full graphics management for producing windows, pie and pie charts, circles and other geometric shapes with Turbo Pascal.

Turbo Graphics (MS-DOS) \$54.95

TEXT EDITING

Punctuation & Style: Improves your writing by catching unbalanced quotes, parentheses and brackets, improper abbreviations, capitalization, sentence structure, much more. It's like having your own copy editor!

Punctuation & Style (CP/M & MS-DOS) \$125.00

Word Finder: This powerful 90,000 word Thesaurus allows you to select the best word for the application. Works inside WordStar for greater ease of use. Instantly searches its dictionary, then displays synonyms, and automatically deletes the "wrong" word and replaces it with the "right" word. Requires 380K disk storage.

Word Finder (CP/M) \$79.95

Wordpatch: Print files with tiny, compressed, wide, or wide compressed type faces, 5 sizes of italic, real superscripts and subscripts, and 6, 7, and 8 lines per inch spacing. No new print controls to learn. Supports most popular dot matrix printers. A must for WordStar users!

Wordpatch (CP/M & MS-DOS) \$49.95

The Word Plus: The ultimate spelling checker. Not only finds misspelled words but shows you correct spelling options, shows the word in context, allows you to build dictionaries of special words you use, and much more.

The Word Plus (MS-DOS) \$150.00

HARDWARE & SUPPLIES

Finger Print "Letter Writer": Unleash your Epson FX series printer. Add near-letter-quality print, IBM and/or Apple Graphics printer emulation, plus 16 other print functions! Three replacement chips quickly fit inside Epson FX series printers. Easy installation. Does not void printer warranty.

Finger Print "LetterWriter" \$79.95

Diskettes, Double Density:

Maxell 10-pack w/ storage box:	3M box of 10:
Single Sided \$19.95	Single Sided \$22.95
Double Sided \$23.95	Double Sided \$26.95

Economy Diskettes: package of 25 including tyvek sleeves.

Single Sided \$29.50
Double Sided \$31.25

Call or write for our FREE catalog

All items are warranted for 90 days. 30 day money back guarantee if not completely satisfied. Guarantee for software applies only if diskette seal is intact. Visa and MasterCard are welcome. Please add 2.00 freight per total order and 2.00 for COD orders. California residents please add 6% sales tax. Prices, availability and specifications subject to change without notice.

CALL TODAY

National

California

(800) 821-8778

(800) 521-7182

Hours: Mon - Fri 8 am - 5 pm PDT

DEALER INQUIRIES WELCOME.



**advent
products inc.**

3154-F E. La Palma Ave.
Anaheim, CA 92806
(714) 630-0446

LISP

FOR THE IBM PERSONAL COMPUTER.

THE PREMIER LANGUAGE
OF ARTIFICIAL
INTELLIGENCE FOR
YOUR IBM PC.

DATA TYPES

Lists and Symbols
Unlimited Precision Integers
Floating Point Numbers
Character Strings
Multidimensional Arrays
Files
Machine Language Code

MEMORY MANAGEMENT

Full Memory Space Supported
Dynamic Allocation
Compacting Garbage Collector

FUNCTION TYPES

EXPR/FEXPR/MACRO
Machine Language Primitives
Over 190 Primitive Functions

IO SUPPORT

Multiple Display Windows
Cursor Control
All Function Keys Supported
Read and Splice Macros
Disk Files

POWERFUL ERROR RECOVERY

8087 SUPPORT

COLOR GRAPHICS

LISP LIBRARY

Structured Programming Macros
Editor and Formatter
Package Support
Debugging Functions
.OBJ File Loader

RUNS UNDER PC-DOS 1.1 or 2.0

IQLISP

5¼" Diskette
and Manual \$175.00

Integral Quality

P.O. Box 31970
Seattle, Washington 98103-0070
(206) 527-2918

Washington State residents add sales tax.
VISA and MASTERCARD accepted.
Shipping included for prepaid orders.

ProDos Copy Listing

(Listing continued, text begins on page 54)

```

61D8:E5 FE      311      SBC      COUNT
61DA:8D D8 BE   312      STA      RWDATA+1      ;Save buffer start address
61DD:8D A4 62   313      STA      WR2BUF+1
61E0:A9 00      314      LDA      $S00      ;Buffer starts on a page boundary
61E2:8D D7 BE   315      STA      RWDATA
61E5:8D A3 62   316      STA      WR2BUF
61E8:           317      *
61E8:           318      *
61E8:           319      *
61E8:AD 6E BE   320      LDA      VPATH2      ;Pathname pointer
61E8:8D 9C G2   321      STA      OPEN2PATH
61E8:AD 6F BE   322      LDA      VPATH2+1
61F1:8D 9D 62   323      STA      OPEN2PATH+1
61F4:20 00 BF   324      JSR      MLI      ;Perform MLI call
61F7:C8         325      DFB      $C8      ;MLI OPEN command code
61F8:9B 62      326      BOC      OPEN2PARM
61FA:90 08 6204 327      BOC      OPEN2SRC      ;No errors, so open the source file
61FC:48         328      PHA      ;Save error code
61FD:20 F8 BE   329      JSR      FRELBUF      ;Release buffer
6200:68         330      PLA      ;Restore error code
6201:4C 8B BE   331      JMP      BADCALL      ;Translate error code and exit
6204:           332      *
6204:           333      *
6204:           334      *
6204:           335      *
6204:A5 74      335      OPEN2SRC LDA      HINEM+1      ;Get buffer start
6206:8D CF BE   336      STA      OSYSBUF+1      ;use general purpose buffer)
6209:A9 00      337      LDA      $S00      ;buffer always on a page boundary
620B:8D CE BE   338      STA      OSYSBUF
620E:A9 C8      339      LDA      $C8      ;MLI open code
6210:20 70 BE   340      JSR      GOSYSTEM      ;Perform the MLI call
6213:B0 1D 6232 341      BCS      ERRORRET      ;Exit if error detected
6215:AD D0 BE   342      LDA      OREFNUM      ;Set up file reference number
6218:8D D2 BE   343      STA      NEWLINEP      ;For NEWLINE call
621D:8D D6 BE   344      STA      RWRREFNUM      ;For READ call
621E:8D DE BE   345      STA      CREFNUM      ;For CLOSE call
6221:A9 00      346      LDA      $S00      ;Set up NEWLINE call
6223:8D D3 BE   347      STA      NLINENBL      ;to disable NEWLINE char
6226:A9 0D      348      LDA      $D0      ;Non-zero
6228:8D D4 BE   349      STA      NLINENBL+1
622B:A9 C9      350      LDA      $C9      ;MLI NEWLINE code
622D:20 70 BE   351      JSR      GOSYSTEM      ;Perform the MLI call
6230:90 07 6239 352      BOC      TRANSFER      ;If no errors detected
6232:48         353      PHA      ;Save error code
6233:20 65 62   354      JSR      CLEANUP      ;Close files and free buffer
6236:68         355      PLA      ;Restore error code
6237:38         356      SEC      ;Flag error
6238:60         357      RTS      ;Return to system
6239:           358      *
6239:           359      *
6239:           360      *
6239:AD A0 62   361      TRANSFER LDA      OPEN2REF      ;Set up WRITE file reference number
623C:8D A2 62   362      RELOC18 STA      WR2REF
623F:A9 CA      363      XFERLOOP LDA      $CA      ;MLI READ code
6241:20 70 BE   364      JSR      GOSYSTEM      ;Perform the MLI call
6244:90 06 624C 365      BOC      XFENOK      ;No errors
6246:C9 05      366      CMP      $S05      ;End of file?
6248:F0 1B 6265 367      BEQ      CLEANUP      ;Yes, so clean up and exit
624A:D0 E6 6232 368      BNE      ERRORRET      ;Exit with appropriate error
624F:8D DB BE   369      XFEROK  LDA      RWRTRANS      ;Number of bytes transferred
624F:8D A5 62   370      RELOC19 STA      WR2RDQ      ;Request to write these bytes
6252:AD DC BE   371      LDA      RWRTRANS+1
6255:8D A6 62   372      RELOC20 STA      WR2RDQ+1
6258:20 00 BF   373      JSR      MLI      ;Write the data to the destination file
625B:CB         374      DFB      $CB      ;MLI WRITE code
625C:A1 62      375      RELOC21 DW      WR2PARMS
625E:90 DF 623F 376      BOC      XFERLOOP
6260:20 8B BE   377      JSR      BADCALL      ;Translate to BI error code
6263:B0 CD 6232 378      BCS      ERRORRET      ;Always
6265:           379      *
6265:           380      *
6265:           381      *
6265:           382      *
6265:A9 00      382      CLEANUP LDA      $S00      ;Default to no errors detected
6267:85 FC      383      STA      TEMP
6269:A9 CC      384      LDA      $CC      ;MLI CLOSE code
626B:20 70 BE   385      JSR      GOSYSTEM      ;Close source file
626E:90 02 6272 386      BOC      RELOC22      ;If no error detected
6270:85 FC      387      STA      TEMP      ;Update error
6272:AD A0 62   388      RELOC22 LDA      OPEN2REF      ;Close destination file
6275:8D A4 62   389      RELOC23 STA      CLOSE2REF
6278:20 00 BF   390      JSR      MLI      ;Perform the MLI call
627B:CC         391      DFB      $CC      ;MLI CLOSE code
627C:A9 62      392      RELOC24 DW      CL2PARMS      ;CLOSE parameter list
627E:90 05 6285 393      BOC      FREE      ;If no error detected
6280:20 8B BE   394      JSR      BADCALL      ;Translate to BI error
6283:85 FC      395      STA      TEMP      ;Update error
6285:20 F8 BE   396      FREE    JSR      FRELBUF      ;Deallocate file buffer
6288:18         397      CLC      ;Default to no errors detected
6289:A5 FC      398      LDA      TEMP      ;Error code in A
628B:F0 01 628E 399      BEQ      RET      ;Zero if no errors
628D:38         400      SEC      ;Flag error
628E:60         401      RTS
628F:           402      *
628F:           403      *
628F:           404      *
628F:07         405      *
6290:           406      *
6290:0002       406      CRPARMS DFB      $07      ;Parameter count
6292:           407      CRPATH  DS      2      ;Pointer to pathname
6293:           408      CRACCESS DS      1      ;Access permitted byte
6294:           409      CRFILEID DS      1      ;File type
6296:01         410      CRFILEID DS      2      ;File aux type
6297:00 00      411      CRDATE  DFB      $00,$00      ;Standard file (Not directory)
6299:00 00      412      CRTIME   DFB      $00,$00      ;Create date
629B:           413      *
629B:           414      *
629B:           415      *
629B:           416      *
629B:           417      *
629B:           418      *
629B:           419      *
629B:           420      *
629B:           421      *
629B:           422      *
629B:           423      *
629B:           424      *
629B:           425      *
629B:           426      *
629B:           427      *
629B:           428      *
629B:           429      *
629B:           430      *
629B:           431      *
629B:           432      *
629B:           433      *
629B:           434      *
629B:           435      *
629B:           436      *
629B:           437      *
629B:           438      *
629B:           439      *
629B:           440      *
629B:           441      *
629B:           442      *
629B:           443      *
629B:           444      *
629B:           445      *
629B:           446      *
629B:           447      *
629B:           448      *
629B:           449      *
629B:           450      *
629B:           451      *
629B:           452      *
629B:           453      *
629B:           454      *
629B:           455      *
629B:           456      *
629B:           457      *
629B:           458      *
629B:           459      *
629B:           460      *
629B:           461      *
629B:           462      *
629B:           463      *
629B:           464      *
629B:           465      *
629B:           466      *
629B:           467      *
629B:           468      *
629B:           469      *
629B:           470      *
629B:           471      *
629B:           472      *
629B:           473      *
629B:           474      *
629B:           475      *
629B:           476      *
629B:           477      *
629B:           478      *
629B:           479      *
629B:           480      *
629B:           481      *
629B:           482      *
629B:           483      *
629B:           484      *
629B:           485      *
629B:           486      *
629B:           487      *
629B:           488      *
629B:           489      *
629B:           490      *
629B:           491      *
629B:           492      *
629B:           493      *
629B:           494      *
629B:           495      *
629B:           496      *
629B:           497      *
629B:           498      *
629B:           499      *
629B:           500      *
629B:           501      *
629B:           502      *
629B:           503      *
629B:           504      *
629B:           505      *
629B:           506      *
629B:           507      *
629B:           508      *
629B:           509      *
629B:           510      *
629B:           511      *
629B:           512      *
629B:           513      *
629B:           514      *
629B:           515      *
629B:           516      *
629B:           517      *
629B:           518      *
629B:           519      *
629B:           520      *
629B:           521      *
629B:           522      *
629B:           523      *
629B:           524      *
629B:           525      *
629B:           526      *
629B:           527      *
629B:           528      *
629B:           529      *
629B:           530      *
629B:           531      *
629B:           532      *
629B:           533      *
629B:           534      *
629B:           535      *
629B:           536      *
629B:           537      *
629B:           538      *
629B:           539      *
629B:           540      *
629B:           541      *
629B:           542      *
629B:           543      *
629B:           544      *
629B:           545      *
629B:           546      *
629B:           547      *
629B:           548      *
629B:           549      *
629B:           550      *
629B:           551      *
629B:           552      *
629B:           553      *
629B:           554      *
629B:           555      *
629B:           556      *
629B:           557      *
629B:           558      *
629B:           559      *
629B:           560      *
629B:           561      *
629B:           562      *
629B:           563      *
629B:           564      *
629B:           565      *
629B:           566      *
629B:           567      *
629B:           568      *
629B:           569      *
629B:           570      *
629B:           571      *
629B:           572      *
629B:           573      *
629B:           574      *
629B:           575      *
629B:           576      *
629B:           577      *
629B:           578      *
629B:           579      *
629B:           580      *
629B:           581      *
629B:           582      *
629B:           583      *
629B:           584      *
629B:           585      *
629B:           586      *
629B:           587      *
629B:           588      *
629B:           589      *
629B:           590      *
629B:           591      *
629B:           592      *
629B:           593      *
629B:           594      *
629B:           595      *
629B:           596      *
629B:           597      *
629B:           598      *
629B:           599      *
629B:           600      *
629B:           601      *
629B:           602      *
629B:           603      *
629B:           604      *
629B:           605      *
629B:           606      *
629B:           607      *
629B:           608      *
629B:           609      *
629B:           610      *
629B:           611      *
629B:           612      *
629B:           613      *
629B:           614      *
629B:           615      *
629B:           616      *
629B:           617      *
629B:           618      *
629B:           619      *
629B:           620      *
629B:           621      *
629B:           622      *
629B:           623      *
629B:           624      *
629B:           625      *
629B:           626      *
629B:           627      *
629B:           628      *
629B:           629      *
629B:           630      *
629B:           631      *
629B:           632      *
629B:           633      *
629B:           634      *
629B:           635      *
629B:           636      *
629B:           637      *
629B:           638      *
629B:           639      *
629B:           640      *
629B:           641      *
629B:           642      *
629B:           643      *
629B:           644      *
629B:           645      *
629B:           646      *
629B:           647      *
629B:           648      *
629B:           649      *
629B:           650      *
629B:           651      *
629B:           652      *
629B:           653      *
629B:           654      *
629B:           655      *
629B:           656      *
629B:           657      *
629B:           658      *
629B:           659      *
629B:           660      *
629B:           661      *
629B:           662      *
629B:           663      *
629B:           664      *
629B:           665      *
629B:           666      *
629B:           667      *
629B:           668      *
629B:           669      *
629B:           670      *
629B:           671      *
629B:           672      *
629B:           673      *
629B:           674      *
629B:           675      *
629B:           676      *
629B:           677      *
629B:           678      *
629B:           679      *
629B:           680      *
629B:           681      *
629B:           682      *
629B:           683      *
629B:           684      *
629B:           685      *
629B:           686      *
629B:           687      *
629B:           688      *
629B:           689      *
629B:           690      *
629B:           691      *
629B:           692      *
629B:           693      *
629B:           694      *
629B:           695      *
629B:           696      *
629B:           697      *
629B:           698      *
629B:           699      *
629B:           700      *
629B:           701      *
629B:           702      *
629B:           703      *
629B:           704      *
629B:           705      *
629B:           706      *
629B:           707      *
629B:           708      *
629B:           709      *
629B:           710      *
629B:           711      *
629B:           712      *
629B:           713      *
629B:           714      *
629B:           715      *
629B:           716      *
629B:           717      *
629B:           718      *
629B:           719      *
629B:           720      *
629B:           721      *
629B:           722      *
629B:           723      *
629B:           724      *
629B:           725      *
629B:           726      *
629B:           727      *
629B:           728      *
629B:           729      *
629B:           730      *
629B:           731      *
629B:           732      *
629B:           733      *
629B:           734      *
629B:           735      *
629B:           736      *
629B:           737      *
629B:           738      *
629B:           739      *
629B:           740      *
629B:           741      *
629B:           742      *
629B:           743      *
629B:           744      *
629B:           745      *
629B:           746      *
629B:           747      *
629B:           748      *
629B:           749      *
629B:           750      *
629B:           751      *
629B:           752      *
629B:           753      *
629B:           754      *
629B:           755      *
629B:           756      *
629B:           757      *
629B:           758      *
629B:           759      *
629B:           760      *
629B:           761      *
629B:           762      *
629B:           763      *
629B:           764      *
629B:           765      *
629B:           766      *
629B:           767      *
629B:           768      *
629B:           769      *
629B:           770      *
629B:           771      *
629B:           772      *
629B:           773      *
629B:           774      *
629B:           775      *
629B:           776      *
629B:           777      *
629B:           778      *
629B:           779      *
629B:           780      *
629B:           781      *
629B:           782      *
629B:           783      *
629B:           784      *
629B:           785      *
629B:           786      *
629B:           787      *
629B:           788      *
629B:           789      *
629B:           790      *
629B:           791      *
629B:           792      *
629B:           793      *
629B:           794      *
629B:           795      *
629B:           796      *
629B:           797      *
629B:           798      *
629B:           799      *
629B:           800      *
629B:           801      *
629B:           802      *
629B:           803      *
629B:           804      *
629B:           805      *
629B:           806      *
629B:           807      *
629B:           808      *
629B:           809      *
629B:           810      *
629B:           811      *
629B:           812      *
629B:           813      *
629B:           814      *
629B:           815      *
629B:           816      *
629B:           817      *
629B:           818      *
629B:           819      *
629B:           820      *
629B:           821      *
629B:           822      *
629B:           823      *
629B:           824      *
629B:           825      *
629B:           826      *
629B:           827      *
629B:           828      *
629B:           829      *
629B:           830      *
629B:           831      *
629B:           832      *
629B:           833      *
629B:           834      *
629B:           835      *
629B:           836      *
629B:           837      *
629B:           838      *
629B:           839      *
629B:           840      *
629B:           841      *
629B:           842      *
629B:           843      *
629B:           844      *
629B:           845      *
629B:           846      *
629B:           847      *
629B:           848      *
629B:           849      *
629B:           850      *
629B:           851      *
629B:           852      *
629B:           853      *
629B:           854      *
629B:           855      *
629B:           856      *
629B:           857      *
629B:           858      *
629B:           859      *
629B:           860      *
629B:           861      *
629B:           862      *
629B:           863      *
629B:           864      *
629B:           865      *
629B:           866      *
629B:           867      *
629B:           868      *
629B:           869      *
629B:           870      *
629B:           871      *
629B:           872      *
629B:           873      *
629B:           874      *
629B:           875      *
629B:           876      *
629B:           877      *
629B:           878      *
629B:           879      *
629B:           880      *
629B:           881      *
629B:           882      *
629B:           883      *
629B:           884      *
629B:           885      *
629B:           886      *
629B:           887      *
629B:           888      *
629B:           889      *
629B:           890      *
629B:           891      *
629B:           892      *
629B:           893      *
629B:           894      *
629B:           895      *
629B:           896      *
629B:           897      *
629B:           898      *
629B:           899      *
629B:           900      *
629B:           901      *
629B:           902      *
629B:           903      *
629B:           904      *
629B:           905      *
629B:           906      *
629B:           907      *
629B:           908      *
629B:           909      *
629B:           910      *
629B:           911      *
629B:           912      *
629B:           913      *
629B:           914      *
629B:           915      *
629B:           916      *
629B:           917      *
629B:           918      *
629B:           919      *
629B:           920      *
629B:           921      *
629B:           922      *
629B:           923      *
629B:           924      *
629B:           925      *
629B:           926      *
629B:           927      *
629B:           928      *
629B:           929      *
629B:           930      *
629B:           931      *
629B:           932      *
629B:           933      *
629B:           934      *
629B:           935      *
629B:           936      *
629B:           937      *
629B:           938      *
629B:           939      *
629B:           940      *
629B:           941      *
629B:           942      *
629B:           943      *
629B:           944      *
629B:           945      *
629B:           946      *
629B:           947      *
629B:           948      *
629B:           949      *
629B:           950      *
629B:           951      *
629B:           952      *
629B:           953      *
629B:           954      *
629B:           955      *
629B:           956      *
629B:           957      *
629B:           958      *
629B:           959      *
629B:           960      *
629B:           961      *
629B:           962      *
629B:           963      *
629B:           964      *
629B:           965      *
629B:           966      *
629B:           967      *
629B:           968      *
629B:           969      *
629B:           970      *
629B:           971      *
629B:           972      *
629B:           973      *
629B:           974      *
629B:           975      *
629B:           976      *
629B:           977      *
629B:           978      *
629B:           979      *
629B:           980      *
629B:           981      *
629B:           982      *
629B:           983      *
629B:           984      *
629B:           985      *
629B:           986      *
629B:           987      *
629B:           988      *
629B:           989      *
629B:           990      *
629B:           991      *
629B:           992      *
629B:           993      *
629B:           994      *
629B:           995      *
629B:           996      *
629B:           997      *
629B:           998      *
629B:           999      *
629B:          1000      *

```

(Continued on page 68)

Tools That Make Your Job Easier

For PC DOS/MSDOS (2.0 and above/128K) • IBM PC/Compatibles, PC Jr., Tandy 1000/1200/2000, & others
For CPM80 2.2/3.0 (Z80 required/64K) • 8" SSSD, Kaypro 2/4, Osborne I SD/DD, Apple II, & others

MIX EDITOR

Programmable, Full/Split
Screen Text Processor

Introductory
Offer

29⁹⁵

Great For All Languages

A general purpose text processor, the MIX Editor is packed with features that make it useful with any language. It has auto indent for structured languages like Pascal or C. It has automatic line numbering for BASIC (255 character lines). It even has fill and justify for English.

Split Screen

You can split the screen horizontally or vertically and edit two files simultaneously.

Custom Key Layouts

Commands are mapped to keys just like WordStar. If you don't like the WordStar layout, it's easy to change it. Any key can be mapped to any command. You can also define a key to generate a string of characters, great for entering keywords.

Macro Commands

The MIX Editor allows a sequence of commands to be executed with a single keystroke. You can define a complete editing operation and perform it at the touch of a key.

Custom Setup Files

Custom keyboard layouts and macro commands can be saved in setup files. You can create a different setup file for each language you use.

MSDOS Features

Execute any DOS command or run another program from inside the editor. You can even enter DOS and then return to the editor by typing exit.

MIX C COMPILER

Full K&R Standard C Language
Unix Compatible Function Library

Introductory
Offer

39⁹⁵

Complete & Standard

MIX C is a complete and standard implementation of C as defined by Kernighan and Ritchie. Coupled with a Unix compatible function library, it greatly enhances your ability to write portable programs.

The Best C Manual

MIX C is complemented by a 400 page manual that includes a tutorial. It explains all the various features of the C language. You may find it more helpful than many of the books written about C.

Fast Development

MIX C includes a fast single pass compiler and an equally fast linker. Both are executed with a simple one line command. Together they make program development a quick and easy process.

Fast Execution

The programs developed with MIX C are fast. For example, the often quoted prime number benchmark executes in a very respectable 17 seconds on a standard IBM PC.

Standard Functions

In addition to the functions described by K&R, MIX C includes the more exotic functions like *setjmp* and *longjmp*. Source code is also included.

Special Functions

MIX C provides access to your machine's specific features through BDOS and BIOS functions. The CHAIN function lets you chain from one program to another. The MSDOS version even has one function that executes any DOS command string while another executes programs and returns.

Language Features

- Data Types: char, short, int, unsigned, long, float, double (MSDOS version performs BCD arithmetic on float and double-no roundoff errors)
- Data Classes: auto, static, extern, register
- Struct, Union, Bit Fields (struct assignment supported)
- Typedef, Initialization
- All operators and macro commands are supported

30 DAY MONEY BACK GUARANTEE

Orders Only: Call Toll Free 1-800-523-9520, (Texas only 1-800-622-4070)

MIX Editor ____ \$29.95 + shipping (\$5 USA/\$10 Foreign) Texas residents add 6% sales tax

MIX C ____ \$39.95 + shipping (\$5 USA/\$25 Foreign) Texas residents add 6% sales tax

Visa ____ MasterCard ____ Card # _____ Exp. Date _____

COD ____ Check ____ Money Order ____ Disk Format _____

Computer _____ Operating System: MSDOS ____ PC DOS ____ CPM80 ____

Name _____

Street _____

City/State/Zip _____

Country _____

Phone _____

MSDOS is a trademark of Microsoft PC DOS is a trademark of IBM CPM80 is a trademark of Digital Research WordStar is a trademark of MicroPro

MIX 2116 E. Arapaho
Suite 363
Richardson, Tx 75081
software
Dealer Inquiries Welcome
Call (214) 783-6001

Instant-C:TM The Fastest Interpreter for C

Runs your programs 50 to 500 times faster than any other C language interpreter.

Any C interpreter can save you compile and link time when developing your programs. But only **Instant-C** saves your time by running your program at compiled-code speed.

Fastest Development. A program that runs in one second when compiled with an optimizing compiler runs in two or three seconds with **Instant-C**. Other interpreters will run the same program in two minutes. Or even ten minutes. Don't trade slow compiling and linking for slow testing and debugging. *Only Instant-C will let you edit, test, and debug at the fastest possible speeds.*

Fastest Testing. **Instant-C** immediately executes any C expression, statement, or function call, and display the results. Learn C, or test your programs faster than ever before.

Fastest Debugging. **Instant-C** gives you the best source-level debugger for C. Single-step by source statement, or set any number of conditional breakpoints throughout your program. Errors always show the source statements involved. Once you find the problem, test the correction in seconds.

Fastest Programming. **Instant-C** can directly generate executable files, supports full K & R standard C, comes with complete library source, and works under PC-DOS, MS-DOS, or CP/M-86.

Instant-C gives you working, well-tested programs faster than any other programming tool. Satisfaction guaranteed, or your money back in first 31 days. **Instant-C** is \$495.

Rational
Systems, Inc.

P.O. Box 480
Natick, MA 01760
(617) 653-6194

ProDos Copy Listing

(Listing continued, text begins on page 54)

```
629B:03      416 OPEN2PARM DFB $03      ;Parameter count
629C:      0002 417 OPEN2PATH DS 2      ;Pointer to pathname
629E:      0002 418 OPEN2BUF DS 2      ;Pointer to I/O buffer
62A0:      0001 419 OPEN2REF DS 1      ;File reference number
62A1:      420 *-----
62A1:      421 * Parameter list for WRITE
62A1:      422 *-----
62A1:04      423 WR2PARMS DFB $04      ;Parameter count
62A2:      0001 424 WR2REF DS 1      ;File reference number
62A3:      0002 425 WR2BUF DS 2      ;File transfer buffer
62A5:      0002 426 WR2REQ DS 2      ;Number of bytes to transfer
62A7:      0002 427 WR2TRANS DS 2      ;Number of bytes transferred
62A9:      428 *-----
62A9:      429 * Parameter list for CLOSE
62A9:      430 *-----
62A9:01      431 CL2PARMS DFB $01      ;Parameter count
62AA:      0001 432 CLOSE2REF DS 1      ;File reference number
62AB:      62AB 433 END EQU *
```

```
3D A1H      3C A1L      3F A2H      3E A2L
43 A4H      42 A4L      BEBB BADCALL      6083 BITMAPS1
6073 BITMAPS      BF58 BITMAP      BEDE CFREFNUM      62A9 CL2PARMS
6265 CLEANUP      62AA CLOSE2REF      6109 CMDSTRU      6104 CMDSTRU
FE COUNT      6292 CRACCESS      6294 CRAUXID      ?6297 CDATE
?6196 CREATE      6293 CRFILID      628F CRPARMS      6290 CRPATH
?6299 CRTIME      62AB END      6232 ERRRET      BE09 ERROUT
BE06 EXTRNCMD      BE56 FBITS      ?BEB7 FIACCESS      BE09 FIAUXID
BEBB FIFILID      6176 FILEOK      60C8 FILLPAGE      BEF8 FREQBUF
6285 FREE      BEF5 GETBUF      61B8 GETMORE      BE70 GOSYSTEM
61CC GOTENOUGH      600B GOSPACE      73 HIMEM      ?0200 INBUF
6100 LINK      6172 MISMATCH      BEF00 MLI      6159 MLIERR
FE2C MOVE      BED2 NEWLREF      BED3 NLINENBL      611A NKTCHR
612A NKTCHR2      FD OFFSET      629E OPEN2BUF      629B OPEN2PARM
629C OPEN2PATH      62A0 OPEN2REF      6204 OPENSRC      BE0D ORFNUM
BECE OSYSBUF      6103 PAGES      615B PARMSOK      BE54 PBITS
600F PROTECT      FE PTR      61AC RELOC10      61B1 RELOC11
61D4 RELOC12      61E5 RELOC14      611D RELOC1      61DD RELOC13
61EB RELOC15      61F1 RELOC16      61F8 RELOC17      623C RELOC18
624F RELOC19      6255 RELOC20      625C RELOC21      6272 RELOC22
6275 RELOC23      6122 RELOC2      627C RELOC24      613D RELOC3
617C RELOC4      6182 RELOC5      6187 RELOC6      61BD RELOC7
6193 RELOC8      619A RELOC9      6042 RELOCATE      6091 RELTABLE
628E RET      615A RETURN      BEF8 RSHIMEM      BE09 RSCOUNT
BED7 RWDATA      BEDE RWREFNUM      61A1 SUCCESS      614E SECON
BE84 SSGINFO      610E START      61A1 SUCCESS      6157 SYNEROR
60C7 TABLESIZE      FC TEMP      61B4 TRANSBUF      6239 TRANSFER
BE5C VPAT11      BE6E VPAT12      62A3 WR2BUF      62A1 WR2PARMS
62A2 WR2REF      62A5 WR2REQ      62A7 WR2TRANS      BE53 XCNUM
623F XFERLOOP      624C XFEROK      BE52 XLIN      BE50 XTRNADDR
```

```
3C A1L      3D A1H      3E A2L      3F A2H
42 A4L      43 A4H      73 HIMEM      FC TEMP
FD OFFSET      FE PTR      FE COUNT      ?0200 INBUF
600B GOSPACE      600F PROTECT      6042 RELOCATE      6073 BITMAPS
6083 BITMAPS1      6091 RELTABLE      60C7 TABLESIZE      60C8 FILLPAGE
6100 LINK      6103 PAGES      6104 CMDSTRU      6109 CMDSTRU
610E START      611A NKTCHR      611D RELOC1      6122 RELOC2
612A NKTCHR2      613D RELOC3      614E SECON      6157 SYNEROR
6159 MLIERR      615A RETURN      615B PARMSOK      6172 MISMATCH
6176 FILEOK      617C RELOC4      6182 RELOC5      6187 RELOC6
618D RELOC7      6193 RELOC8      6196 CREATE      619A RELOC9
61A1 SUCCESS      61AC RELOC10      61B1 RELOC11      61B4 TRANSBUF
61B8 GETMORE      61CC GOTENOUGH      61DD RELOC12      61DD RELOC13
61E5 RELOC14      61EB RELOC15      61F1 RELOC16      61F8 RELOC17
6204 OPENSRC      6232 ERRRET      6239 TRANSFER      623C RELOC18
623F XFERLOOP      624C XFEROK      624F RELOC19      6255 RELOC20
625C RELOC21      6265 CLEANUP      6272 RELOC22      6275 RELOC23
627C RELOC24      6285 FREE      628E RET      628F CRPARMS
6290 CRPATH      6292 CRACCESS      6293 CRFILID      6294 CRAUXID
6297 CDATE      6299 CRTIME      629B OPEN2PARM      629C OPEN2PATH
629E OPEN2BUF      62A0 OPEN2REF      62A1 WR2PARMS      62A2 WR2REF
62A3 WR2BUF      62A5 WR2REQ      62A7 WR2TRANS      62A9 CL2PARMS
62AA CLOSE2REF      62AB END      BE06 EXTRNCMD      BE09 ERROUT
BE50 XTRNADDR      BE52 XLIN      BE53 XCNUM      BE54 PBITS
BE56 FBITS      BE6C VPAT11      BE6E VPAT12      BE70 GOSYSTEM
BEBB BADCALL      BE14 SSGINFO      ?BEB7 FIACCESS      BEB8 FIFILID
BEB9 FIAUXID      BECE OSYSBUF      BE0D ORFNUM      BEB2 NEWLREF
BED3 NLINENBL      BEDE RWREFNUM      BE09 ERROUT      BEF8 FREQBUF
BED4 RWDATA      BEDE CFREFNUM      BEF5 GETBUF      BEF8 FREQBUF
BEFB RSHIMEM      BF00 MLI      BF58 BITMAP      FE2C MOVE
```

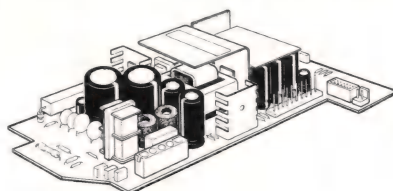
*** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 15-JAN-84 21:28
** TOTAL LINES ASSEMBLED 433
** FREE SPACE PAGE COUNT 79

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

Switching Power Supply!

- + 5VDC - 8 Amps
- +12VDC - 5 Amps
- 12VDC - 1 Amp
- (81 WATTS MAX)



\$29⁹⁵
ea.

4 FOR \$99

ADD \$2
EA. UPS

BRAND NEW UNITS, MFG. BY BOSCHERT FOR HEWLETT PACKARD! PERFECT FOR SMALL COMPUTER AND DISK DRIVE APPLICATIONS #XL81-5630. INPUT 110V/220V, 50/60 HZ. NOMINAL OUTPUTS: +5VDC @ 8A, +12VDC @ 5A, -12VDC @ 1A. TOTAL MAX OUTPUT. MUST BE LIMITED TO 81 WATTS TOTAL! (WITH PIN OUT SHEET.) ORIGINAL FACTORY BOXED.

64K S100 STATIC RAM

\$119⁰⁰
KIT

LOW POWER!
150 NS ADD \$10

BLANK PC BOARD
WITH DOCUMENTATION
\$49.95

SUPPORT ICs + CAPS
\$17.50

FULL SOCKET SET
\$14.50

FULLY SUPPORTS THE
NEW IEEE 696 S100
STANDARD
(AS PROPOSED)

FOR 56K KIT \$105

ASSEMBLED AND
TESTED ADD \$50

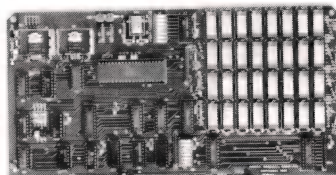


FEATURES: PRICE CUT!

- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports IEEE 696 24 BIT Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- 2716 EPROMs may be installed in any of top 48K.
- Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- BOARD may be partially populated as 56K.

256K S-100 SOLID STATE DISK SIMULATOR!
WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



BLANK PCB
(WITH CP/M* 2.2
PATCHES AND INSTALL
PROGRAM ON DISKETTE)
\$69⁹⁵
(8203-1 INTEL \$29.95)

- FEATURES:
- 256K on board, using + 5V 64K DRAMS.
 - Uses new Intel 8203-1 LSI Memory Controller.
 - Requires only 4 Dip Switch Selectable I/O Ports.
 - Runs on 8080 or Z80 S100 machines.
 - Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
 - Provisions for Battery back-up.
 - Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
 - The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
 - Compare our price! You could pay up to 3 times as much for similar boards.

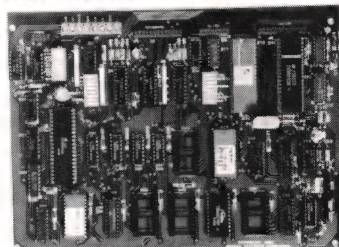
\$149⁰⁰
(ADD \$50 FOR A&T)
#LS-100 (FULL 256K KIT)

THE NEW ZRT-80 CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- RS232 at 16 BAUD Rates from 75 to 19,200.
- 24 x 80 standard format (60 Hz).
- Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- Higher density formats require up to 3 additional 2K x 8 6116 RAMs.
- Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- Composite or Split Video.
- Any polarity of video or sync.
- Inverse Video Capability.
- Small Size: 6.5 x 9 inches.
- Upper & lower case with descenders.
- 7 x 9 Character Matrix.
- Requires Par. ASCII keyboard.



\$89⁹⁵
#ZRT-80
(COMPLETE KIT, 2K VIDEO RAM)

BLANK PCB WITH 2716
CHAR. ROM. 2732 MON. ROM
\$49⁹⁵

SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

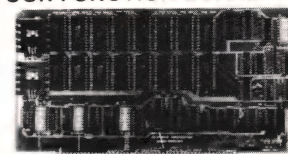
FOR 8 IN. SOURCE DISK
(CP/M COMPATIBLE)
ADD \$10

NEW!

EPROM II
FULL
EPROM KIT
\$69.95
A&T EPROM
ADD \$35.00

32K S100 EPROM/STATIC RAM

FOUR FUNCTION BOARD!



NEW!

BLANK
PC BOARD
WITH DATA
\$39.95
SUPPORT
IC'S
PLUS CAPS
\$16
FULL
SOCKET SET
\$15

We took our very popular 32K S100 EPROM Card and added additional logic to create a more versatile EPROM/RAM Board.

FEATURES:

- This one board can be used in any one of four ways:
 - As a 32K 2716 EPROM Board
 - As a 32K 2732 EPROM Board (Using Every Other Socket)
 - As a mixed 32K 2716 EPROM/2K x 8 RAM Board
 - As a 32K Static RAM Board
- Uses New 2K x 8 (TMM2016 or HM6116) RAM's
- Fully Supports IEEE 696 Buss Standard (As Proposed)
- Supports 24 Bit Extended Addressing
- 200 NS (FAST!) RAM's are standard on the RAM Kit
- Supports both Cromemco and North Star Bank Select
- Supports Phantom
- On Board wait State Generator
- Every 2K Block may be disabled
- Addressed as two separate 16K Blocks on any 64K Boundary
- Perfect for MP/M* Systems
- RAM Kit is very low power (300 MA typical)

PRICES
SLASHED!

32K STATIC RAM KIT — \$99.95

For RAM Kit A&T — Add \$40

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

CP/M-68K Conditional Batch Processing

by Roger E. Donais

Anyone who uses batch processing is certain to recognize the value of conditional instructions. A few basic submit statements, such as SKIPIF, PAUSEIF, and ABORTIF have been added to CP/M-80 in the past. These instructions make it possible for compiling, assembling, and linking to be handled by a single submit file that is smart enough to stop the process if any phase should fail.

CP/M-68K and CP/M-80 are similar in many ways. In fact, files and programs can share the same disk without adverse consequences. They are, however, as different as the processors they support. Because CP/M-80 creates a working copy of a submit file, that file becomes the natural target for implementing batch control

programming effort by taking advantage of the parsing done by the CCP, and yet present a recognizable form of conditional statement.

IF <parameter> <condition>
.<action> /\$\$\$\$/ <statement>

Condition: The first four characters taken from the file-name field of the file control block created by the CCP are used to identify the condition to be tested. An asterisk may be appended to indicate that the complementary condition is to be used.

AMBIG tests for an ambiguous parameter (wildcards).
EXISTS tests for an existing file.
EMPTY tests for an empty file or nonexistent file.

A simple "if" can greatly speed the software development process.

functions. On the other hand, CP/M-68K stores the expanded file in memory. The fact is that a direct approach would require more knowledge about the undocumented aspects of CP/M-68K than can be acquired in an evening!

IF.68K (Listing One, page 72) began as two outlines. The first listed what was wanted and the other described how it would be accomplished. One pencil later the following compromise was reached:

Syntax: The conditional statement is structured so as to minimize pro-

NULL tests for an undefined parameter.

Actions: The first character taken from the file-type field of the same file control block is used to identify the desired action. An asterisk may again be appended to indicate that the complementary action is to be taken.

S(kip) ignores the statement following /\$\$\$\$/.
Q(uit) aborts submit file processing.
P(ause) prompts for operator assistance.

The double-slash quad-dollar-sign token separates the conditional statement from the executable CP/M com-

Roger E. Donais, 7506 Republic Ct., Alexandria, VA 22306

mand tail. When the submit file is expanded, a dollar sign is used as the leading character to signal formal parameter replacement. If the next character is not an ASCII digit, the character is written to the expanded output and no replacement is performed. This simply means that we use four dollar signs in the submit file but have the program search for only two.

One disadvantage with this compromise approach is that CP/M-68K will pass only one command. This limits the range of the conditional statement to one CP/M-68K command. It's bad enough that the same conditional statement has to be entered over and over again in order to control a range of commands, but the aggravation of watching it reload each time is just too much!

With a little camouflage, CP/M can be tricked into passing more than one command. The best character seems to be a semicolon. Very few programs dare use this character because it marks the beginning of a comment. This makes it a prime candidate for an alternate command separator. It affects very few programs and conceals its true intent from CP/M, and our program need only replace these bogus separators with the exclamation marks that CP/M-68K expects to find.

The problem with this wonderful logic is that CP/M-68K seems to stop building the command tail after it finds a semicolon. Because the basic technique could use any character, a backslash became the next choice. Although the entire command tail is passed, the chain function does not protect the referenced line but leaves it where it is. It doesn't take much imagination to realize what happens when the default buffer is overwritten. So we are limited to using it with built-in commands and the aggravation of using only one external command.

Let's look at the resulting kludge. The program starts by looking for an action character in the second file control block. If it fails to find a match, as would be the case with a null parameter, it flips a switch before testing the first file control block. Control is then transferred ac-

ording to the indicated action.

Each action begins by making a call to evaluate the condition. This evaluation ends with a jump that complements the original condition according to a NOT condition and a NOT action request. This is definitely a kludge that capitalizes upon the parsing and setup done by the CCP. The file-name and file-type fields are

normally padded with spaces (an even \$20). The presence of an asterisk (CP/M wildcard) changes the padding character to a question mark (an odd \$3F). The code uses this even/odd fill character to blindly obtain the desired state.

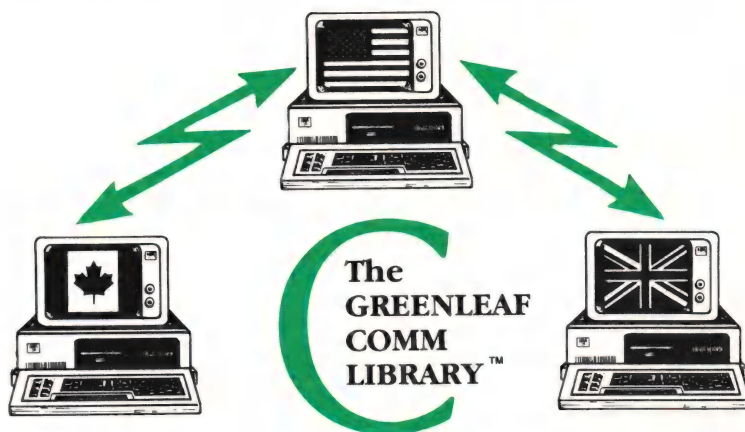
The two complementary labels, TRUE and FALSE, are truly illusory. What began as a simple matter

ARE YOU TRYING TO COMMUNICATE ?

C programs can communicate with the world now through the power of **The Greenleaf Comm Library**. Now from the people who brought you **The Greenleaf Functions** General Library for C, comes this rich interrupt driven, ring-buffered asynchronous communications capability.

Over 100 functions in C and assembler to facilitate communications at up to 9600 baud. Up to eight ports at a time. ASCII or XMODEM. X-On/X-Off too. Hayes compatible modems controlled here. Safe too, bet you can't exit your application with interrupts hot. Major applications around the world base their communicating applications on **The Greenleaf Comm Library**. Stop just trying and start really communicating. Get your copy of **The Greenleaf Comm Library** today. For all major C compilers, all models, all versions. For the IBM PC and just about any machine with MSDOS and an 8086. Comes with source code, extensive examples, demo programs, featuring **C-Terminal**, reference card and newsletter. No royalty. \$185

Other Products: **The Greenleaf Functions** General Library, over 220 functions for total control of the IBM PC, with source. \$185 for the compilers listed below. (See ordering instructions below).



Specify compiler when ordering: Lattice, Microsoft, Computer Innovations, Mark Williams, or DeSmet. Add \$7.00 for UPS Second Day Air (or \$5.00 for ground). Texas residents add sales tax. Mastercard, VISA, check, or P.O. In stock, shipped same day.

<input type="checkbox"/> General Libraries	\$185
<input type="checkbox"/> Comm Library	\$185
<input type="checkbox"/> C186 Compiler	\$349
<input type="checkbox"/> Lattice C	\$395
<input type="checkbox"/> Mark Williams	\$475

For Information:
214/446-8641



PRICES ARE SUBJECT TO
CHANGE WITHOUT NOTICE.

2101 HICKORY DR.
CARROLLTON, TX 75006

Circle no. 97 on reader service card.

of going to one or the other, based upon the resulting condition, turned into quite a nightmare. At times it seemed that neither one was right. Had they been ZIP and ZAP they might not have confused the issue so much. Debugging eventually reached a point where I would no longer look at them or at the remarks. It became, "Well, it's BEQ now, so let's try BNE" or "target is TRUE, so try FALSE!" It became worse regardless of what I tried. Finally, everything was wrong. That made it easy. TRUE became FALSE and FALSE became TRUE. I no longer care how it looks nor if it makes sense. It works and that is the end of it!

A skip is performed by merely ending the program and returning to the CCP. The command is executed by invoking the CP/M-68K chain function after replacing all semicolons

with exclamation marks.

There may be a more direct way to abort than the method used. Programs, however, evolve according to a programmer's knowledge and imagination. My 68000 and CP/M-68K knowledge is very limited, so that left only imagination. The *CP/M User's Guide* says that a submit statement contained in a submit file would transfer control if the referenced file exists. Otherwise the statement would be ignored and processing would resume with the next statement in the current file.

The solution is obvious: invoke an existing, empty submit file—*et voilà!* The actual code goes one step more. It creates a submit file with a single instruction that subsequently erases itself. The result may not be fast, but it does abort without having to keep more junk on the disk.

Yes, the whole thing has the potential to be fooled and foiled. But if you remain within the guidelines and don't wander beyond its incomplete parsing and blind processing, it performs exactly as intended. The crazy little submit file in Listing Two (page 82) may well be the world's slowest ERAQ, but it does provide an example to demonstrate the performance of IF.68K. The additional example in Listing Three (page 82) not only conditionally assembles and links the modules of an assembly project but also copies the changed source files from a RAM disk to floppy. Naturally, execution speed is improved by loading IF.68K from a RAM disk rather than floppy.

DDJ

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 195.

CP/M-68K (Text begins on page 70)

Listing One

```
*****
*
* PROGRAM:  IF.68K                      LAST UPDATE:  10 Mar 85
*
* PURPOSE:  PROVIDE BASIC CONTROL FOR CP/M-68k SUBMIT FACILITY
*
* AUTHOR:   Roger E. Donais            TEL: (703) 765-0615
*           7506 Republic Ct.
*           Alexandria, VA 22306
*
* UPDATE LOG:
*
* 1.2   10 Mar 85 --- Replaced non-working semi-colon with
*                   back-slash as command separator and
*                   added ^Z check for empty file ...red
* 1.1   14 Nov 84 --- Added semi-colon as multiple command
*                   separator ...red
* 1.0   11 Jul 84
*
*****
*
* SYNTAX:
*
* IF <Filename> <Condition>.<Action> /$$$$/ <statement>
*
* CONDITION IS: AMBIG - TO TEST FOR AMBIGIOUS FILESPEC
*               EXIST - TO TEST FOR AN EXISTING FILE
*               EMPTY - TO TEST FOR AN EMPTY FILE
*               NULL - TO TEST FOR A NULL PARAMETER
*
* ACTION IS:    S - TO SKIP STATMENT PART OF COMMAND LINE
*               Q - TO QUIT SUBMIT FILE PROCESS
*               P - TO PAUSE FOR OPERATOR ASSISTANCE
*
* STATEMENT IS ANY VALID CP/M COMMAND
*
* NOTE:  AN ASTERISK (*) MAY BE APPENDED TO THE CONDITION
*        AND/OR THE ACTION TO INDICATE NEGATION.  AMBIG*
*        WOULD THUS TRANSLATE "NOT AMBIGIOUS" AND Q* WOULD
*        TRANSLATE "DO NOT QUIT".
*
```

(Continued on page 74)

A FULL C COMPILER FOR \$49.95

Limited time offer:
**FREE CED
TEXT EDITOR
WITH EACH ECO-C88**

Ecosoft's Eco-C88 C Compiler

An unbeatable value! For \$49.95 you set:

- A C compiler with all data types and operators (except bit fields)
- Fast executing code. Some common benchmark results are:

	Eco-C88	L (1)	C86 (1)	MS (1)	MW (1)
sieve	12	11	13	11	12
fib	43	58	46	109	—
deref	14	13	—	10	11
matrix	22	29	27	28	29

1. Computer Language, Feb., 1985. Reproduced with permission.

- 8087 support using a single library. The 8087 is sensed at runtime and used if present.
- A standard library with over 200 functions (many of which are System V compatible for greater portability).
- Error messages in English - no cryptic numbers to look up.
- A cc and "mini-make" (in source) that makes compiling a snap.
- ASM or OBJ output (for use with MSDOS linker).
- Expanded user's manual.
- Works with all IBM PC's and clones using MSDOS 2.1 or later.

ORDERS ONLY

1-800-952-0472

The ECOSOFT family of C products ORDER FORM

- ☐ C Compiler \$49.95 _____
- ☐ Programming Guide \$20.00 _____
- ☐ Self-Study Guide \$17.00 _____
- ☐ Programmer's Library \$20.00 _____
- ☐ Program Editor \$29.95 _____
- ☐ C Library Source \$10.00 _____
- ☐ ISAM (.OBJ) \$15.00 _____

Total* (Ind. res. add 5% tax) _____

*Please add \$4.00 for shipping.

Payment: VISA MC AE Check

Credit card expir. date _____

Card # _____

Name _____

Address _____

City, state _____

Zip _____ Phone _____

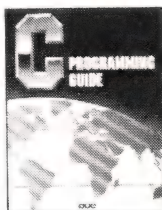
Ecosoft, Inc.
6413 N. College Ave.
Indianapolis, IN 46220
(317) 255-6476 • 8:30-4:30



Other Eco-C Products

CED Program Editor. \$29.95

A screen-oriented program text editor similar to the Turbo Pascal editor. You can create, compile and link the source file with CED. If there is an error, CED automatically reloads the source file and places the cursor on the offending section of code. CED supports editing multiple files (with windows), macros, more than 50 editing commands and is configurable to your particular needs and preferences. An outstanding value.

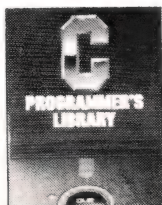


C Programming Guide, 2nd Ed. (Purdum, Que Corp.) \$19.95

This best seller walks you through the C language in an easy-to-read manner. All aspects of the language are covered, including many of the new ANSI Standards suggestions. Many of the error messages issued by the Eco-C88 compiler reference page numbers in this text making an ideal learning environment.

C Self-Study Guide (Purdum, Que Corp.) \$16.95

This new book is designed for the person that is learning C on their own. The book is filled with questions and answers that most beginning C programmers have. It also includes many sample programs that illustrate tips, traps and techniques that may take years to discover otherwise. A perfect complement to the Guide book.



C Programmer's Library (Purdum, Leslie, Stegemoller, Que Corp.) \$19.95

This best seller is an intermediate level text designed to show you how to write functions for your C library. The book contains many useful library additions, including an ISAM file handler, plus sections on advanced C topics.

TRADEMARKS: ECO-C88, ECOSOFT, TURBO PASCAL, BORLAND INT'L.

Listing One

```

*
*      MULTIPLE COMMANDS MAY BE SEPARATED BY SEMI-COLONS, AN
*      EXCLAMATION MARK (!) OR END OF LINE ENDS STATEMENT.
*
*****

* CP/M FUNCTION DEFINITIONS
bdos          equ 2
fnABORT       equ 0
fnCINP        equ 1
fnCOUT        equ 2
fnPRINT       equ 9
fnRESET       equ 13
fnOPEN        equ 15
fnCLOSE       equ 16
fnDELETE      equ 19
fnREAD        equ 20
fnWRITE       equ 21
fnCREATE      equ 22
fnDMA         equ 26
fnCHAIN       equ 47

.page
* MISC CONSTANTS
FCB1          equ $5C          1st CP/M File Control Block
FCB2          equ $38          2nd CP/M File Control Block
NAM1          equ FCB1+1      1st FCB File Name field
NAM2          equ FCB2+1      2nd FCB File Name field
TYP1          equ NAM1+8      1st FCB File Type field
TYP2          equ NAM2+8      2nd FCB File Type field
BUFF          equ $80         Default CP/M Buffer / Command Line
SIZE          equ $0F+FCB1    CP/M File Size (Sector Count) FOR FCB1

*****
* PHASE-1 --- PROCESS (P)AUSE), (S)KIP and (Q)UIT PARAMETERS

      MOVE.L 4(A7),A6          A6 Contains Base Page
      LEA 128(A6),A3          DEFAULT BUFFER
      MOVE.B TYP2(A6),D0      Get Action Character - 2nd FCB

RETRY:
      CMP.B #'S',D0
      BEQ SKIP
      CMP.B #'Q',D0
      BEQ QUIT
      CMP.B #'P',D0
      BEQ PAUSE
      MOVE.B TYP1(A6),D0      Get Action Character - 1st FCB
      EOR #1,NULLFLG
      BNE RETRY

ERROR:
      MOVE.L #MESSAGE,A1

ERROR1:
      MOVE.B (A1)+,D1
      BEQ ERROR2
      EXT D1
      MOVE #fnCOUT,D0
      TRAP #bdos
      BRA ERROR1

ERROR2:
      MOVE #fnABORT,D0
      TRAP #bdos

NULLFLG:
      DC.W 0000

.page
*****
* PHASE-2 --- PROCESS AMBIG, EXISTS, EMPTY and NULL PARAMETERS

COMPARE:
      MOVE #4-1,D0          4 Character token count
      LEA NAM2(A6),A0      Get Token address
      TST NULLFLG

```



```

BEQ L1
LEA NAM1(A6),A0
L1:  LSL.L #8,D2          Make room for next character
      MOVE.B (A0)+,D2     and load it
      DBRA D0,L1          Repeat for all 4-characters

      MOVE.L #TABLE,A0    Point to Function Table
      MOVE #4-1,D0        Number of Entries
L2:  MOVE.L (A0)+,D1       Get Current Entry
      CMP.L D2,D1
      BEQ JUMP            Exit on Match
      ADDA #4,A0          Else Step to Next Entry
      DBRA D0,L2          and Repeat for ALL Entries
L3:  ADDA #4,A7            Pop callers from stack
      BRA ERROR          and abort
JUMP:
      TST NULLFLG
      BEQ JMP1
      TST D0
      BNE L3
JMP1: MOVE.L (A0),A0      EVERYTHING GOOD ON 2ND FCB
      JMP (A0)            AND ONLY NULL ON 1ST FCB
                          JUMP TO FUNCTION

```

.page

SKIP: * SKIP COMMAND TAIL IF CONDITION TRUE

BSR COMPARE

BNE CHAIN

Execute Command Tail if FALSE
and Ignore if TRUE

EXIT: RTS

QUIT: * QUIT (ABORT) SUBMIT IF CONDITION TRUE

BSR COMPARE

BNE CHAIN

Ignore Command Tail if FALSE

ABORT:

MOVE.L #ABORTMSG,D1

MOVE #fnPRINT,D0

TRAP #bdos

(Continued on next page)

“**XTREE**™ has redefined my relationship with my hard drive.”

—PC MAGAZINE

Why XTREE is the New Standard For File and Directory Management.

XTREE simplifies file and directory handling by providing single keystroke commands to access, delete, rename, view, move, list or show all files within any and all directories. XTREE displays a graphic picture of your directory organization, instantly shows all the files in each directory or all files across all directories! Just point with the arrow keys (full scrolling and paging in all windows) and press a key. Your menu is clearly displayed at all times.

10 Reasons To Buy XTREE

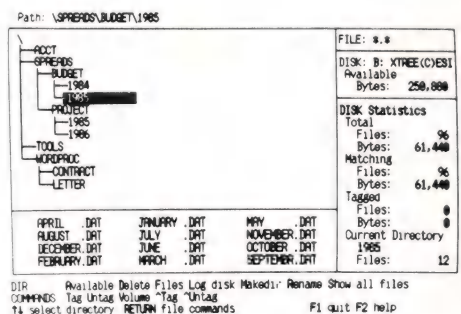
1. Shows **ALL** files or groups of files in **ALL** directories in one sorted display.
2. Copy, delete or rename multiple files in different directories in **ONE** operation.
3. Automatically copies groups of files across several diskettes.
4. **NOT** copy-protected.
5. Demo available for only \$5.00 includes reusable diskette case (holds up to 5 disks).
6. “I finally found a package that I use routinely to clean up my disks. I recommend it highly. It's a steal at \$49.95.”
— John Dvorak, Infoworld
7. “All XTREE operations execute very quickly. XTREE is a great utility for getting an overall feel of your hard disk. And, if you feel the need to totally reorganize your hard disk, XTREE is unparalleled.”
— Garry Ray, PC Week

8. “If the program were not already subtitled *The new standard for file and directory management*, I would have described this wonderful new utility the exact same way.”
— Phil Wiswell, PC Magazine

9. Only \$49.95 plus \$2 shipping.



10. To Order call 800-634-5545 In CA (818) 990-3457
or send check or money order to:
EXECUTIVE SYSTEMS, INC.
15300 Ventura Blvd., Suite 305, Sherman Oaks, CA 91403



Circle no. 180 on reader service card.

Listing One

```
MOVE    #fnRESET,D0
TRAP    #bdos           Make Certain Drive A: is R/W
MOVE.L  #ABORTREC,D1
MOVE    #fnDMA,D0       Setup for File Output
TRAP    #bdos
MOVE.L  #ABORTFCB,D1
MOVE    #fnDELETE,D0    Delete any '$$$ABORT.SUB' File
TRAP    #bdos
MOVE    #fnCREATE,D0     Then make a new one
TRAP    #bdos
MOVE    #fnWRITE,D0      Fill it
TRAP    #bdos
MOVE    #fnCLOSE,D0      Close it
TRAP    #bdos
MOVE.L  #ABORTSUB,D1     and Chain to it
BRA     CHAIN1
```

PAUSE: * PAUSE IF CONDITION TRUE

BSR COMPARE

BNE CHAIN

EXECUTE W/O STOPPING IF FALSE

X9:

MOVE.L #PROMPT,D1

MOVE #fnPRINT,D0

TRAP #bdos

MOVE #fnCINP,D0

(Continued on page 78)

68020

Cross Assembler

Features:

- Instruction Set Fully Compatible with Motorola (Including 68000/08/10 and 68020 Instructions)
- User Defined Symbols Unique to 31 Characters
- 32 Bit Addressing Supporting All Motorola Addressing Modes
- Powerful Macro Facility
- Structured Directives
- Symbol Cross Reference Listings
- Conditional Assembly Directives
- Linking Loader
- Motorola Standard Hexadecimal S-Record Format
- Object Module Output
- Librarian
- Conversion and Download Utilities Supporting Tektronix and Hewlett-Packard

Now available: 68020 Symbolic Debugger
In 4th quarter '85: 68020 C-Cross Compiler

1930 Freeway Circle, Suite 101, Santa Clara, CA 95051
Mailing Address: P.O. Box 60337, Sunnyvale, CA 94088

 **MICROTEC[®]**
RESEARCH

The Microtec ASM68K Cross Assembler System is a Microtec Research implementation of the assembly language specified by Motorola Corporation for its 68000/68010/68020 microprocessor family. ASM68K is used to assemble and link/load programs on a general purpose host computer, then download those programs for execution on the above microprocessor. The Microtec ASM68K package includes macro assembler, linking loader, Tektronix conversion utility, and download utility. An optional object module librarian is also available which enhances the effectiveness of the linking loader.

Quality, Compatibility and Service are the differences which set Microtec Research apart from others.

If you're a serious software developer, shopping for software development tools, call or write today for more information:

800-551-5554.

In CA call (408) 733-2919.

DE SMET C

8086/8088 Development Package

\$109

FULL DEVELOPMENT PACKAGE

- Full K&R C Compiler
- Assembler, Linker & Librarian
- Full Screen Editor
- Execution Profiler
- Complete STDIO Library (>120 Func)

Automatic DOS 1.X/2.X SUPPORT

BOTH 8087 & S/W FLOATING POINT

OVERLAYS

OUTSTANDING PERFORMANCE

- First and Second in AUG '83 BYTE benchmarks

SYMBOLIC DEBUGGER \$50

- Examine & change variables by name using C expressions
- Flip between debug and display screen
- Display C source during execution
- Set multiple breakpoints by function or line number

DOS LINK SUPPORT \$35

- Converts DeSmet.O to DOS.OBJ Format
- LINKs with DOS ASM
- Uses Lattice* naming conventions

CWARE

CORPORATION

P.O. Box C, Sunnyvale, CA 94087

(408) 720-9696

Street Address: 505 W. Olive, #767 (94086) Call for hrs.

All orders shipped UPS surface on IBM format disks. Shipping included in price. California residents add sales tax. Canada shipping add \$5, elsewhere add \$15. Checks must be on U.S. Bank and in U.S. Dollars. Call 9am-1pm to CHARGE by VISA/MC/AMEX.

Foreign Distributors: AFRICA: HI-TECH SVCS, Gaborone 4540 or Telex 2205BD LANGER • ENGLAND: MLH Tech, 0606-891146 • JAPAN: JSE 03-486-7151 • SWEDEN: ESCORT DATA 08-87 41 48 or THESEUS KONSULT 08-23 61 60



Transform Your Programs with CPP—C Preprocessor Plus

Includes ALL features of the standard C preprocessor.

- Define arbitrarily complex macros with #define command.
- Include and nest files to any depth with #include command.
- Conditionally include or exclude lines with #if, #ifdef and #ifndef commands.
- Optional extra feature: Imbed formatting or other commands in your source code. (Lines starting with . or * are ignored.)

Fast and flexible

- 30 times faster than the Preprocessor published in Dr. Dobb's Journal.
- Can be used for any language, including assembler.
- Can be used as a stand-alone macro/include processor.
- Code can be used as the lexical analyzer for parsers or assemblers.

Complete

- You get complete SOURCE CODE in standard C.
- You get everything you need to use CPP immediately.
- CPP is unconditionally guaranteed. If for any reason you are not satisfied with CPP, your money will be refunded promptly.

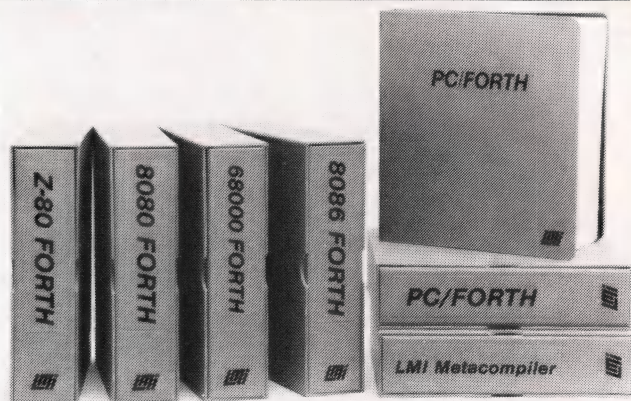
Price: \$50.

Call or write today:
Edward K. Ream
1850 Summit Ave., Dept. DD
Madison, WI 53705
(608) 231-2952

TO ORDER: Specify both the operating system (MS-DOS, CP/M 80 or CP/M 68K) and the disk format (8 inch CP/M or the exact type of 5 1/4 inch disk). Send a check or money order for \$50 (\$60 for foreign orders). Foreign checks must be denominated in U.S. dollars drawn on a U.S. bank. Sorry, I do NOT accept phone, credit card or COD orders. Please do NOT send purchase orders unless a check is included.

Circle no. 90 on reader service card.

TOTAL CONTROL with LMI FORTH™



For Programming Professionals:
an expanding family of compatible, high-performance, Forth-83 Standard compilers for microcomputers

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compile to 8080, Z-80, 8086, 68000, and 6502
- No license fee or royalty for compiled applications

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

Call or write for detailed product information and prices. Consulting and Educational Services available by special arrangement.



Laboratory Microsystems Incorporated

Post Office Box 10430, Marina del Rey, CA 90295

Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, D-7820 Titisee-Neustadt

UK: System Science Ltd., London EC1A 9JX

France: Micro-Sigma S.A.R.L., 75008 Paris

Japan: Southern Pacific Ltd., Yokohama 220

Australia: Wave-onic Associates, 6107 Wilson, W.A.

Circle no. 66 on reader service card.

Listing One

```

TRAP #bdos
AND #5F,D0          Force upper case
CMP.B #'Q',D0       Quit on "Q"
BEQ ABORT
CMP.B #'S',D0       Skip on "S"
BEQ EXIT
CMP.B #'C',D0       And continue on "C"
BNE X9

.page
CHAIN: * CHAIN TO COMMAND TAIL

      ADD #BUFF,A6    Point to command line
      MOVE.B (A6)+,D0  Get command length
      EXT D0

X5:   DBRA D0,X6       Ignore Statement if Command
      RTS              Tail cannot be found

X6:   CMP.B #'/',(A6)+ Look for leading slash
      BNE X5

      CMP.B #' ',-2(A6) Insure that the previous
      BEQ X7            character was either
      CMP.B #9,-2(A6)   a space or tab
      BNE X5

X7:   CMP.B #'$',(A6)+ The Original four dollar
      BNE X5            signs will appear
      CMP.B #'$',(A6)+ as only two, so
      BNE X5            check for 2 $'s

      CMP.B #'/',(A6)+ Check for trailing slash
      BNE X5

      CMP.B #' ',(A6)+ And finally delimiting
      BEQ X8            space or tab
      CMP.B #9,-1(A6)
      BNE X5

X8:   * So we got the command tail. Let's take a quick break
      * and repalce all semicolons with exclamation marks.

      move.l A6,A1      copy pointer and
      move D0,D1        character count,
      bra Y10           then begin search

Y9:   move.b #'!',-1(A1)

Y10:  cmp.b #'\\',(A1)+ search until eol or '\\'
      dbeq D1,Y10       substitute & continue if '\\'
      beq Y9

      SUB #1,A6          Step back to new len position
      ADD #1,D0          adjust remaining command length
      MOVE.B D0,(A6)     and punch it
      MOVE.L A6,D1       Put the command tail in D1

CHAIN1: MOVE #fnDMA,D0    Make addr in D1
      TRAP #bdos         the Default Buffer
      MOVE #fnCHAIN,D0   and Chain it
      TRAP #bdos

*****
OPEN: MOVE.L A6,D1        Base page to D1
      ADD.L #FCB1,D1     and step to fcb
      MOVE #fnOPEN,D0
      TRAP #bdos
      CMP.B #5F,D0       Set flags on open file
      RTS

.page
NULL1: CMP.B #520,NAM1(A6) Test for no file name
      BNE TRUE

```

(Continued on page 80)

GRAF 3.0

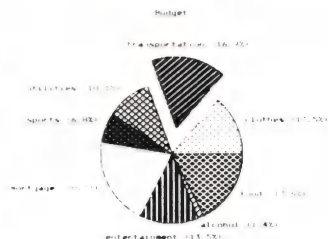
the complete
BUSINESS and SCIENTIFIC
printer graphics program

\$ 49⁹⁵

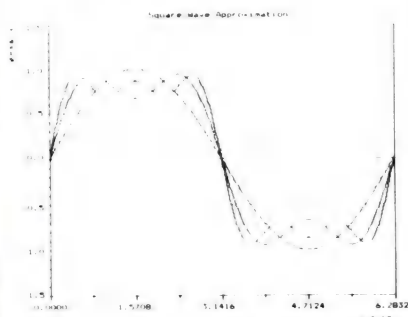
\$ 69⁹⁵

CP/M-80

MS-DOS / PC-DOS



- display floating point data directly from spreadsheets, data bases, word processors, and programming languages (or the keyboard) in a wide variety of bar, pie, line, and scatter plots
- plot and group up to 6 different variables on a single graph, distinguished by up to 14 different fill-in patterns and 8 different point-plotting symbols
- menu driven operation supports automatic graph scaling, labeling, and legend creation
- add up to 5 different-density grid lines, and choose from a wide variety of numerical labeling options



GRAF 2.0 Update Policy: Returning your original GRAF 2.0 disk to MSC entitles you to \$20.00 off the above prices.

Numeric Compressor Option: 8087 support for the MS-DOS/PC-DOS version of GRAF 3.0 is available for an additional \$20.00.

TERMS: We ship via first class mail. The above prices include domestic shipping and handling. Orders outside USA require additional \$5.00 for postage. N.Y. residents add state sales tax. When ordering you MUST state your computer and printer make and model. We support MS-DOS (PC-DOS) version 2.0 or later on computers with at least 192k RAM, and CP/M-80 version 2.2 or later on Z80 computers (other than modified Apples) supporting a TPA of at least 54k (requires 64k of RAM). Most soft-sector disk formats are available. (If you can read several formats, please send us a list.) GRAF 3.0 works with any printer fully compatible with one of the following: Epson FX, RX, LX, MX (with GRAFTRAX), or LQ-1500; C. Itoh Prowriter; NEC 8023A; Star Micronics Gemini 10X, 15X, SG-10, SG-15; IBM Graphics Printer; Okidata 192; and earlier Okidata models equipped with the "IBM Plug 'n' Play" chips. (If you have an Okidata printer, other than the 192, the Plug 'n' Play chips are required!)

MSC

Microcomputer
Systems
Consultants

27 Forest Avenue
Port Jefferson Station
New York 11776-1820

Circle no. 136 on reader service card.

BD Software, Inc., maker of the original
CP/M-80 C Language Development
System, knows

Time is precious

So the compilation, linkage and execution speeds of BDS C are the fastest available, even (especially!) on floppy-based systems. Just ask any user! With 15,000+ packages sold since 1979, there are *lots* of users . . .

New! Ed Ream's RED text editor has been integrated into the package, making BDS C a truly complete, self-contained C development system.

Powerful original features: CDB symbolic source-level debugger, fully customizable library and run-time package (for convenient ROM-ing of code), XMODEM-compatible telecommunications package, and other sample applications.

National C User's Group provides direct access to the wealth of public-domain software written in BDS C, including text editors and formatters, BBS's, assemblers, C compilers, games and much more.

Complete package price: \$150.

All soft-sectored disk formats, plus Apple CP/M, available off-the-shelf. Shipping: free, by UPS, within USA for *prepaid* orders. Canada: \$5. Other: \$25. VISA, MC, COD, rush orders accepted.

BD Software, Inc.

BD Software, Inc.
P O Box 2368
Cambridge MA 02238
617 • 576 • 3828



Circle no. 123 on reader service card.

Listing One

```
FALSE:  MOVE #0,D0          Start with FALSE
        BRA TOGGLE

TRUE:   MOVE #1,D0          Start with TRUE
TOGGLE: EOR.B D0,TYP2-1(A6)  TOGGLE "NOT CONDITION"
        MOVE.B TYP2-1(A6),D0
        EOR.B D0,TYP2+2(A6)  TOGGLE "NOT ACTION"
        BTST #0,TYP2+2(A6)   FLAG RESULT
        RTS

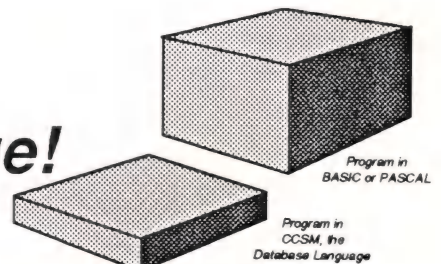
*****
NULL:   * TEST FIRST FCB FOR NULL PARAMETER

        TST NULLFLG         If FCB2 is in use the
        BEQ NULL1           Condition is FCB1 Filename
        EOR.B D0,TYP1-1(A6)  else TOGGLE
        MOVE.B TYP1-1(A6),D0 using FCB1 Pad Characters
        EOR.B D0,TYP1+2(A6)
        BTST #0,TYP1+2(A6)
        RTS

*****
EMPTY:  * TEST FIRST FCB FOR NULL, NON-EXISTANT OR EMPTY FILE

        CMP.B #20,NAM1(A6)   NO FILE NAME IS EMPTY FILE
        BEQ FALSE
        BSR OPEN             NON-EXISTANT IS ALSO EMPTY
        BEQ FALSE
        MOVEQ #fnREAD,D0
        TRAP #2
        TST D0
```

Write 1/3 the code with CCSM, the Database Language! Only \$59.95!



Now, you can dramatically increase your programming productivity, with CCSM, the Database Language.

If you're writing programs in BASIC or PASCAL, you'll appreciate our full-screen editor. You'll also write about 1/3 the amount of code, with this compact, productive language.

Raw Power for Raw Data

CCSM, the Database Language, developed by COMP Computing, is a proprietary version of ANSI Standard MUMPS. MUMPS as a language has been refined and developed for the past 20 years, and is used in corporate America, and by countless thousands around the world, who must manage large and complicated filing systems.

With CCSM, the Database Language, you won't be restricted with line numbers, and different "dialects", as you are with BASIC. CCSM, the Database Language, will transport directly from your microcomputer, to a mini or mainframe WITHOUT MODIFICATION!

CCSM, the Database Language, is also "type-less". That means you can use variables "on the fly", without the tyranny of type declarations such as found in PASCAL.

B-Tree Record Structure, and All the Goodies

CCSM, the Database Language, doesn't make you decide when to use "sequential" files, or "random" files...all records are stored in a B-Tree file for rapid access, and maximum flexibility. In terms of technical excellence, CCSM, the Database Language, utilizes virtual memory, so programs or local variables can expand to the size of the disk. CCSM's sophisticated buffer-pooling provides faster performance than RAM-disk alternatives...8087 and BCD support, included!

Stop Kludging, Start Computing!

Programming and database solutions are as close as your phone. Call Guidance Software now, and get CCSM, the Database Language, for your 128K IBM, or IBM-compatible. (Macintosh version, coming) CCSM, the Database Language, sells for \$59.95, and includes full documentation. For another \$9.95, we'll send a "cookbook" of programs, and a second disk of popular subroutines and utilities. If you need bar charts and pie graphs, also order the graphics disk for \$49.95. A multi-user version (up to 15) is available for \$450. MasterCard, VISA, and AMEX accepted by phone.

1-800-257-8052

In Texas, 713-529-2576

Guidance Software, PO Box 5362, Kingwood, TX 77325

IBM and Macintosh are trademarks of International Business Machines, and Apple.

Circle no. 98 on reader service card.


```

BNE FALSE          NO RECORD == EMPTY FILE
CMP.B #1A, (A3)
BEQ FALSE          ^Z MUST BE EMPTY ASCII FILE
BRA TRUE

```

```

*****
EXIST:  * TEST FIRST FCB FOR EXISTING FILE

```

```

BSR OPEN
BNE FALSE          File not Found Therefore Non-Existant
BRA TRUE          otherwise it's naturally there

```

```

*****
AMBIG:  * TEST FIRST FCB FOR AMBIGIOUS NAME OR TYPE

```

```

      MOVE #11-1,D0          Test 8-Char NAME + 3-Char TYPE
X4:
      CMP.B #'?',NAM1(A6,D0)
      DBEQ D0,X4
      BNE TRUE              Contains "?" Therefore Ambigious
      BRA FALSE

```

```

.page
*****
***** DATA STORAGE AREA *****
TABLE  DC.B 'AMBI'
        DC.L AMBIG          *****
        DC.B 'EMPT'        *           NOTE           *
        DC.L EMPT          *           *               *
        DC.B 'EXIS'        * NULL MUST BE THE LAST TABLE *
        DC.L EXIST         * ENTRY. SEE COMPARE SUBROUTINE *
        DC.B 'NULL'        *           *               *
        DC.L NULL          *****

```

```

ABORTSUB:
        DC.B 10, 'A:$$$$ABORT',0
ABORTREC:
        DC.B 'ERA A:$$$$$ABORT.SUB', $D, $A

```

(Continued on next page)

Time and Money.

We've just done something we know you'll like. We've made the SemiDisk far more affordable than ever before. With price cuts over 25% for most of our product line. Even our new 2 megabyte units are included.

It's Expandable

SemiDisk Systems builds fast disk emulators for more microcomputers than anyone else. S-100, IBM-PC, Epson QX-10, TRS-80 Models II, 12, and 16. You can start with as little as 512K bytes, and later upgrade to 2 megabytes per board...at your own pace, as your needs expand. Up to 8 megabytes per computer, using only four bus slots, max! Software drivers are available for CP/M 80, MS-DOS, ZDOS, TurboDOS, VALDOCS 2, and Cromix. SemiDisk turns good computers into **great** computers.

SEMIDISK

SemiDisk Systems, Inc., P.O. Box GG, Beaverton, Oregon 97075 503-642-3100

Call 503-646-5510 for CBBS/NW, 503-775-4838 for CBBS/PCS, and 503-649-8327 for CBBS/Aloha. all SemiDisk equipped computer bulletin boards (300/1200 baud) SemiDisk, SemiPool trademarks of SemiDisk Systems.

Battery Backup, Too

At 0.7 amps per 2 megabytes, SemiDisk consumes far less power than the competition. And you don't have to worry if the lights go out. The battery backup option gives you 5-10 hours of data protection during a blackout. Nobody else has this important feature. Why risk valuable data?

The Best News

	512K	1Mbyte	2Mbyte
SemiDisk I, S-100	\$695	\$1395	
SemiDisk II, S-100	\$995		\$1995
IBM PC, XT, AT	\$595		\$1795
QX-10	\$595		\$1795
TRS-80 II, 12, 16	\$695		\$1795
Battery Backup Unit	\$150	\$150	\$150

Someday you'll get a SemiDisk.
Until then, you'll just have to....wait.



Circle no. 85 on reader service card.

Listing One

```
DC.L ';'SUBMIT FILE PROCESSING ABORTED','$D,$A,$!A
ABORTFCB:
DC.B 1,'$$$ABORTSUB'
DC.B 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ABORTMSG
DC.B $D,$A,'ABORTING ...'$

PROMPT:
DC.B $7,$D,$A,$D,$A,'SUBMIT FILE PROCESSING PAUSED.','$D,$A,$D,$A

DC.B $7,'PRESS: (Q) TO QUIT, (S) TO SKIP -or- (C) TO CONTINUE $'

MESSAGE:
DC.B 7,$D,$A,$A
DC.B 'IF.68K Ver 1.2 CONDITIONAL SUBMIT COMMAND FORMAT:',$D,$A,$A
DC.B ' IF <filespec> AMBIGx.yx /$$$$/ <command tail>',$D,$A
DC.B ' IF <filespec> EXISTx.yx /$$$$/ <command tail>',$D,$A
DC.B ' IF <filespec> EMPTYx.yx /$$$$/ <command tail>',$D,$A
DC.B ' IF <parameter> NULLx.yx /$$$$/ <command tail>',$D,$A,$A

DC.B ' WHERE: x is an optional "*" to indicate negation',$D,$A
DC.B ' and y is (Q)uit, (S)kip or (P)ause',$D,$A,$A

DC.B ' NOTE: Command tail may consist of multiple commands',$D,$A
DC.B ' separated by backslashes (\) and ends at an',$D,$A
DC.B ' exclamation mark (!) or physical end of line.'',$D,$A,$A

DC.B ' EXAMPLE: IF A:JUNK.TYP /$$$$/ EMPTY,S CMD1\CMD2!CMD3',$D,$A,$A
DC.B ' Neither CMD1 nor CMD2 will execute if A:JUNK.TYP',$D,$A
DC.B ' is empty. Regardless, CMD3 will always execute.'
```

End Listing One

Listing Two

ERAQ.SUB - File Deletion with Query

```
DIR $1
;-----
IF $1 EXISTS.P /$$$$/ ERA $1
IF $2 NULL.Q /$$$$/
SUBMIT ERAQ $2 $3 $4 $5 $6 $7 $8 $9
```

End Listing Two

Listing Three

ASM.SUB - Simple 68K Assemble and Link

```

IF $1 NULL.S /$$$$/ ERA $1.0
IF FILE1.BAK EXISTS.S* /$$$$/ ERA FILE1.0\PIP B:=FILE1.S
IF FILE2.BAK EXISTS.S* /$$$$/ ERA FILE2.0\PIP B:=FILE2.S
IF FILE3.BAK EXISTS.S* /$$$$/ ERA FILE3.0\PIP B:=FILE3.S
IF FILE4.BAK EXISTS.S* /$$$$/ ERA FILE4.0\PIP B:=FILE4.S
ERA *.BAK
IF FILE1.0 EXISTS.S /$$$$/ A:AS68 -S A: -L FILE1.S
IF FILE2.0 EXISTS.S /$$$$/ A:AS68 -S A: -L FILE1.S
IF FILE3.0 EXISTS.S /$$$$/ A:AS68 -S A: -L FILE1.S
IF FILE4.0 EXISTS.S /$$$$/ A:AS68 -S A: -L FILE1.S
A:LO68 -R FILE1.0 FILE2.0 FILE3.0 FILE4.0

```

End Listings

FORTRAN PROGRAMMERS

Discover why
you should be using
F77L

the complete implementation
of the ANSI FORTRAN 77
Standard for the IBM PC and
compatibles.

If you are serious about your
FORTRAN programming, you
should be using F77L.
\$477



**Lahey Computer
Systems, Inc.**

31244 Palos Verdes Drive West, Suite 243
Rancho Palos Verdes, California 90274
(213) 541-1200

Serving the FORTRAN community
since 1969

Circle no. 67 on reader service card.

TOTAL RECALL FOR \$104.95



Now you can retrieve everything
you entered during program
development -- from version
one thru all updates. SRMS
does it with a complete
source utility at an af-
fordable price.

You can retrieve specific
versions of a program,
make changes and rein-
state the source while
recording when, why and
where changes were made.

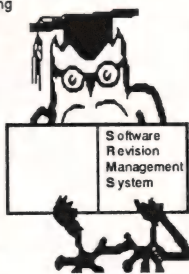
All versions are stored in a
single library without dupli-
cation of common code or
text -- saves disk space. You can also store comments
specific to each version -- provides complete develop-
ment history and documentation of the program.

SRMS supports recall and edit of programs written in
BASIC, FORTRAN, PASCAL (including Turbo Pascal),
C and ASSEMBLY CODE.

Requires MS or PC DOS, Version 2 and up with 128K
and one disk (floppy or hard).

Quilt Computing

7048 Stratford Road
Woodbury, MN 55125
(612) 739-4650



Circle no. 107 on reader service card.

Finally. BSW-Make.

The Boston Software Works now brings a
complete implementation of the Unix
"make" facility to MS-DOS. No more recom-
piling every file in sight after a small edit; no
more wondering if you've really rebuilt every
module affected by an edit. Just type "make"
and BSW-Make automatically builds your
product quickly, efficiently and correctly.

BSW-Make supports:

- most compilers and assemblers
- MS-DOS or PC-DOS v2.00 or later
- macros for parameterized builds
- default rules
- MS-DOS pathnames
- any MS-DOS machine
(192K minimum)

Only \$69.95 postpaid (Mass. residents add 5% sales tax)

The Boston Software Works
120 Fulton Street, Boston, MA 02109
(617) 367-6846

Circle no. 126 on reader service card.

LEARN PROLOG

Interactively and Quickly with PROLOG-86

Ideal for Artificial Intelligence and Prototyping

A complete *Prolog Interpreter, Tutorial, and set of Sample Programs:*

☐ Modify and write Expert Systems.

Use the simple "Guess the animal"
example on the Tutorial or use the
sophisticated system for Section 318 of
the US Tax Code written by one of the
PROLOG-86 authors and published in
the March, 1985 issue of Dr. Dobb's
Journal.

☐ Understand Natural Language

Use the sample program that produces
a dBase DISPLAY command as output.

☐ Write Symbolic Math or Abstract Problem Solving Applications

This is a complete Prolog program to
convert from Fahrenheit to Centigrade:
f_to_c(C,F):- C is(F-32) *5/9. Planning
programs and games are included to
help you learn.

☐ BECOME FAMILIAR WITH PROLOG IN ONE EVENING.

Programming experience is not required, but a logical mind is. Serious development of ex-
perimental systems is practical with PROLOG-86. 1 or 2 pages in Prolog is often equivalent to 10
or 15 in C.

RECENT IMPROVEMENTS: MSDOS commands, on-line help, load Editor.

AVAILABILITY: All MSDOS, PCDOS systems.

ONLY

\$125

Full refund if not
satisfied during
first 30 days.

**Solution
Systems™**

335-D Washington St.
Norwell, Mass. 02061
617-659-1571
800-821-2492

Circle no. 148 on reader service card.

LEARN LISP

Interactively and Write Realistic Programs
with LISP-86 for Only \$75

A "COMMON LISP" compatible Tutorial, Interpreter, Debugging, and
Pretty Printer plus a Fast, Full Screen Editor, Samples and Help

☐ Start Easily and Quickly:

A complete, modular tutorial helps you
learn LISP at your own pace. An in-
tegrated, interactive environment provides
all of the elements needed to enter, modify,
analyze and debug programs.

☐ Natural Language, Expert Systems and Mailing List:

Natural Language concepts are illustrated
by a phone number retrieval program.
Choose the best word processing program
for you with the Expert System. File hand-
ling and typical data processing work are
demonstrated by a Mailing List program.

☐ Write Realistic Programs:

Short examples and substantial programs of
about 10 pages in length help you learn by
modifying, studying and using the key con-
cepts needed to write programs of 1000
lines or more.

☐ The "COMMON LISP" Standard:

LISP-86 includes a 230+ function subset of
the "COMMON LISP" Standard. Use ex-
tras like the MSDOS interface and graphics.
Or use "strict compatibility" to make pro-
grams written in LISP-86, with no changes,
work with other COMMON LISP systems
like VAX LISP, GC/LISP or LISP
Machine LISP.

Use and Modify the "Which Word Processor?" program
to learn about EXPERT SYSTEMS.

Runs on any MSDOS or PCDOS System: Not copy-protected, LISP-86 is available in just
about any 3", 5" or 8" format. PC compatibles can run LISP-86 with no installation pro-
cedure. 192K memory and 1 floppy drive are the minimums required.

ONLY

\$75

Full refund if not
satisfied during
first 30 days.

For Beginners and Experienced Programmers

**Solution
Systems™**

335-B Washington St.
Norwell, Mass. 02061
617-659-1571
800-821-2492

Circle no. 149 on reader service card.

ProDOS and the RAM Disk

by Alfred Steele

I have a Synetix Flashcard with 288K of RAM that I use with Apple's CP/M, Pascal, and DOS 3.3. When I got my card more than two years ago, there was no ProDOS. After using ProDOS several times, I knew that I could not live without my RAM disk, so I decided to roll my own driver. I did several things to speed up the process of writing the driver. I used only 256K of the 288K memory on the Flashcard. The Flashcard uses six banks of memory. Four banks have 64K and the other two have only 16K each. I chose to use only the four 64K banks of memory to save development time and reduce code size. The loss of 32K of storage space is not a problem at this time. With the 256K of RAM in use, I get a RAM disk with 512 blocks of storage space.

disk driver can be installed anywhere in memory, but only three options are given at run time. You may place the driver inside a free space in ProDOS at \$FF00. This space may only be free in ProDOS Version 1.01. Later versions of ProDOS may not support the driver at that address. You may also place the driver under the BASIC interpreter or at \$300 hex page 3. The reason I advise you to install the driver inside of ProDOS is to support MousePaint, which uses page 3 of memory. Information on how to use the interpreter can be found in the *ProDOS Technical Reference Manual* and *Beneath Apple ProDOS*. Both books are really necessary.

The code for the RAM-disk driver is not hard to follow. The RAM-disk driver code that will be relocated to any address starts at label SSDSTART

Using a RAM disk speeds hi-res display by a factor of four.

This is Version 1.3 of the RAM-disk driver (see Listing, page 85). Version 1.0 was larger and faster but not relocatable. It had to be loaded in page 3 of memory. Version 1.1 fixed bugs in the installation program. Version 1.2 is the first version in which the RAM-disk driver code is small enough to fit inside of a free space in ProDOS. The program is completely relocatable anywhere in memory. One side effect of making the driver smaller and relocatable is that it is slower than in the other versions.

Version 1.3 fixes some errors in the installation code and adds more user prompts. This version of the RAM-

and ends at ENDSSD. The routine is only 8E hex (142 decimal) bytes long. The main part of the installation code starts at label MAIN. This is a series of subroutines to check if the RAM disk has already been formatted. If this is a first time installation, the program title and the volume name of the RAM disk are displayed. The driver code is relocated to the address of your choice. The ProDOS active devices are listed and the count is updated. The address of the device driver is placed in the vector table for Slot 5, Drive 1. The RAM-disk volume directory information is written to the RAM disk and the volume bit map is written to disk. Information on the diskette volume can be found in Chapter 4 of *Beneath Apple ProDOS* and Appendix B of the *ProDOS Technical*

Alfred Steele, Box 6296, APO NY 09012

Reference Manual.

The program has been written to be easy to use. If you try to install the driver more than once, the program will display a message that a driver has already been installed and abort. If the driver finds a formatted volume directory header on the RAM disk it will ask you if you want to clear it or not. If you type N, the program will install the driver but won't clear the RAM-disk directory. If you type Y, the program will install the driver and reformat the RAM disk. If any other driver, such as a hard disk driver, is already installed in the active device vector for Slot 5, Drive 1, the program will abort with an error message.

Once the RAM-disk driver is installed you can access the RAM disk using standard ProDOS commands and utilities. You can catalog the RAM disk using either the command CAT /RAM or CAT,S5,D1. If you

want to copy files to the RAM disk use your copy of ProDOS FILER [or see the article "Adding a COPY Command to ProDOS" on page 54 of this issue—ed]. The system intergradation is clean and efficient and provides super fast loading and storing of any type of file.

One benchmark that I ran using the RAM disk consisted of loading high-resolution pictures to the screen. I made up a ProDOS test disk with a short BASIC program to load 16 different hi-res pictures as fast as possible. Using Apple's Disk II it took about 35 seconds to display all 16 pictures. Using the same program and pictures loaded to the RAM disk it only took 8.3 seconds to display all 16 pictures. The program runs about 4.2 times faster using the RAM disk for an effective I/O transfer speed of 15.9K per second vs. the Disk II transfer speed of 3.8K per second.

For those who don't want to type in more than 900 lines of code and comments, I will send you a copy of the source and object code on one Apple ProDOS disk for \$15 postpaid anywhere in the United States. Please send a money order because a personal check is difficult to process in West Germany. To assemble the code, you need a copy of the ProDOS assembler/editor, or you can assemble the code under a DOS 3.3 assembler and move the object code to a ProDOS disk. My address is:

Alfred Steele
Box 6296
APO NY 09012

DDJ

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 196.

Ram Disk Listing (Text begins on page 84)

```

SOURCE  FILE #01 =>/RAMDISK.SOURCE/SSD3
0000: 1 *****
0000: 2 *
0000: 3 * FLASHCARD RAMDISK DRIVER *
0000: 4 *
0000: 5 * VERSION 1.3
0000: 6 * JULY 1985
0000: 7 *
0000: 8 * COPYRIGHT 1985
0000: 9 * ALFRED STEELE
0000: 10 *
0000: 11 * THIS PROGRAM MAY BE COPIED *
0000: 12 * FOR PERSONAL, NON-PROFIT USE *
0000: 13 *
0000: 14 *****
0000: 15 *

0000: 16 LST ON
0000: 17 FILE NAME IS /RAMDISK.SOURCE/SSD3.0
0000: 18 ORG $2000
0000: 19 * TEXT STUFF
0000: 20 *
0000: 21 EOL EQU $00
0000: 22 CR EQU $0D
0000: 23 *
0000: 24 * ADDRESS OF SSD DRIVER
0000: 25 *
0000: 26 TEMPZERO EQU $00
0000: 27 SSDCODE EQU $02
0000: 28 *
0000: 29 * MLI DISK DRIVE ZERO PAGE
0000: 30 *
0000: 31 READ EQU $01
0000: 32 WRITE EQU $02
0000: 33 DIRBLOCKS EQU $06
0000: 34 SAVEBYTES EQU $08
0000: 35 *
0000: 36 * DEVICE DRIVER PARAMETERS
0000: 37 *
0000: 38 COMMAND EQU $42
0000: 39 UNITNUMB EQU $43
0000: 40 BUFFADR EQU $44
0000: 41 BLOCKNUM EQU $46
0000: 42 RAMBANK EQU $48
0000: 43 TEMP EQU $4A
0000: 44 *
0000: 45 * LOW BYTE ADDRESS OF BLOCK
0000: 46 * NUMBERS IN SSD RAM DISK
0000: 47 *
0000: 48 BLOCK1 EQU $D2
0000: 49 BLOCK2 EQU $D3
0000: 50 BLOCK3 EQU $D4
0000: 51 BLOCK4 EQU $D5
0000: 52 BLOCK5 EQU $D6
0000: 53 BLOCK6 EQU $D7
0000: 54 *
0000: 55 * SSD ADDRESS FOR CARD IN SLOT 5
0000: 56 *
0000: 57 SSDLOW EQU $C0D0
0000: 58 SSDHI EQU $C0D1
0000: 59 *
0000: 60 CETBUFF EQU $BEF5
0000: 61 MLI EQU $BF00
0000: 62 DEVICNT EQU $BF31
0000: 63 DEVLST EQU $BF32
0000: 64 DEVDADR51 EQU $BF1A
0000: 65 *

2000: 66 * VALUE OF DEVICE DRIVER POINTER
2000: 67 * IF THE SLOT AND DRIVE IS NOT
2000: 68 * BEING USED.
2000: 69 *
2000: 70 NODEV EQU $D0A2
2000: 71 *
2000: 72 * BI GLOBAL PAGE ADDRESS
2000: 73 *
2000: 74 WARMDOS EQU $BE00
2000: 75 *
2000: 76 * HARDWARE LOCATIONS
2000: 77 *
2000: 78 KEY EQU $C000
2000: 79 STROBE EQU $C010
2000: 80 *
2000: 81 * MONITOR SUBROUTINES
2000: 82 *
2000: 83 HOME EQU $FC58
2000: 84 CROUT EQU $FD8E
2000: 85 COUT EQU $FDED
2000: 86 BELL EQU $FF3A
2000: 87 *
2000: 88 *****
2000: 89 *
2000: 90 JMP MAIN
2000: 91 *
2000: 92 *****
2000: 93 *
2000: 94 * THIS IS THE SSD RAMDISK DRIVER
2000: 95 * CODE THAT IS MOVED TO SSDCODE
2000: 96 * AT THE START OF THIS PROGRAM
2000: 97 * THIS ROUTINE MUST START ON A
2000: 98 * PAGE BOUNDARY. I USE THE DUMMY
2000: 99 * DS STATEMENT BELOW SO THAT THE
2000: 100 * ROUTINE WILL START ON A PAGE
2000: 101 * BOUNDARY. IF YOU CHANGE ANY
2000: 102 * CODE ABOVE BE SURE TO ADJUST
2000: 103 * THE DS STATEMENT BEFORE YOU
2000: 104 * ASSEMBLE AND RUN THE PROGRAM.
2000: 105 *
2000: 106 *
2000: 107 DS $FD
2000: 108 *
2000: 109 *
2000: 110 SSDSTART CLD
2000: 111 *
2000: 112 * SAVE ALL MLI ZERO PAGE LOC
2000: 113 * PLUS TWO NEEDED BY DRIVER
2000: 114 *
2000: 115 LDA $C0D3
2000: 116 LDX $SAVEBYTES
2000: 117 SI LDA COMMAND,X
2000: 118 PHA
2000: 119 DEX
2000: 120 BPL SI
2000: 121 *
2000: 122 * COMPUTE WHICH 64K BLOCK WE
2000: 123 * NEED TO READ OR WRITE TO.
2000: 124 * EACH 64K BLOCK OF RAM HAS 128
2000: 125 * PRODOS 512 BYTES BLOCKS IN IT.
2000: 126 * WE ARE ONLY USING THE 64K
2000: 127 * BANKS SO ONLY BLOCKS:
2000: 128 *
2000: 129 * BLOCK1 BLOCKS 0-127
2000: 130 * BLOCK2 BLOCKS 128-255
2000: 131 * BLOCK4 BLOCKS 256-383
2000: 132 * BLOCK5 BLOCKS 384-511

```

(Continued on next page)


```

210C: 133 * ARE BEING USED. WHEN WE KNOW
210C: 134 * WHAT BANK WE WILL USE WE MOVE
210C: 135 * THAT BANKS LOW BYTE FOR THE
210C: 136 * SSD RAM CARD BANK ADDRESS INTO
210C: 137 * THE LOW-BYTE OF THE ZERO PAGE
210C: 138 * POINTER RAMBANK. THE HI-BYTE
210C: 139 * IS SET TO *C0. IF BLOCKNUM+1
210C: 140 * IS NOT ZERO WE NEED TO ACCESS
210C: 141 * BLOCKS 256-383 OR 384-511.
210C: 142 *
210C: 143 *
210C: 144 COMPUTE LDA BLOCKNUM+1
210C: 145 BNE C2
210C: 146 *
210C: 147 * BLOCKS 0-127 OR 128-255
210C: 148 *
210C: 149 LDA BLOCKNUM
210C: 150 BMI C1
210C: 151 *
210C: 152 * BLOCKS 0-127
210C: 153 *
210C: 154 LDA *BLOCK1
210C: 155 BNE C4
210C: 156 *
210C: 157 * BLOCKS 128-255
210C: 158 *
210C: 159 C1 LDA *BLOCK2
210C: 160 BNE C4
210C: 161 *
210C: 162 * BLOCKS 256-383 OR 384-511
210C: 163 *
210C: 164 C2 LDA BLOCKNUM
210C: 165 BMI C3
210C: 166 *
210C: 167 * BLOCKS 256-383
210C: 168 *
210C: 169 LDA *BLOCK4
210C: 170 BNE C4
210C: 171 *
210C: 172 * BLOCKS 384-511
210C: 173 *
210C: 174 C3 LDA *BLOCK5
210C: 175 STA RAMBANK
210C: 176 *
210C: 177 * SET UP HI-BYTE OF SSD CARD
210C: 178 *
210C: 179 LDA *C0
210C: 180 STA RAMBANK+1
210C: 181 *
210C: 182 * DECODE TYPE OF COMMAND FOR
210C: 183 * RAM DISK DRIVER TO DO.

```

```

212C: 184 *
212C: 185 * LDA COMMAND
212C: 186 STA TEMP
212C: 187 BEQ EXIT
212C: 188 *
212C: 189 * SAVE COMMAND IN Y-REG AND
212C: 190 * MASK OUT BITS 1 AND 0.
212C: 191 * IF A-REG = ZERO AFTER MASK
212C: 192 * THEN THIS IS A VALID READ
212C: 193 * OR WRITE COMMAND.
212C: 194 *
212C: 195 * AND *CFC
212C: 196 BEQ SETBLOCK
212C: 197 *
212C: 198 * ERROR COMMAND HANDLER
212C: 199 * INVALID COMMAND SENT
212C: 200 *
212C: 201 LDA *C27
212C: 202 STA TEMP
212C: 203 BNE EXIT
212C: 204 *
212C: 205 * READ OR WRITE COMMAND. SET UP
212C: 206 * BLOCK NUMBER AND BRANCH TO
212C: 207 * READBYTE OR WRITEBYTE.
212C: 208 *
212C: 209 * EACH 64K BANK HOLDS 128 PRODOS
212C: 210 * 512 BYTE BLOCKS. GET THE LOWER
212C: 211 * 7 BITS AND SHIFT ONCE LEFT.
212C: 212 * THIS WILL MULTIPLY THE NUMBER
212C: 213 * 0-127 TO 0-254 EVEN VALUES
212C: 214 * ONLY. THIS ADJUSTED BLOCK
212C: 215 * NUMBER IS USED AS THE HI-BYTE
212C: 216 * OF THE ADDRESS IN THE 64K BANK
212C: 217 * THE LOW-BYTE IS SET TO ZERO
212C: 218 * AND INC FROM 0 TO *FF.
212C: 219 *
212C: 220 SETBLOCK LDA BLOCKNUM
212C: 221 AND *C7F
212C: 222 ASL
212C: 223 STA BLOCKNUM+1
212C: 224 STA SSDHI
212C: 225 *
212C: 226 * MOVE COMMAND TO A-REG AND
212C: 227 * SET UP LOOP COUNT TO RUN
212C: 228 * THRU READ OR WRITE TWICE.
212C: 229 * EACH LOOP THRU READ OR WRITE
212C: 230 * WILL MOVE 256 BYTES.
212C: 231 *
212C: 232 LDV #0
212C: 233 STY BLOCKNUM

```

DATESTAMPER™ has the answers

Drive B1: 4 files, using 15K 110K FREE 14:01-23 Feb					
-- file	size	created	accessed	modified	
B1: ADDRESS .DAT	5K	22:01-17 Jan	08:30-01 Feb	08:23-01 Feb	
B1: JSMITH .LTR	2K	16:30-24 Dec '84	11:59-10 Feb	16:30-24 Dec '84	
B1: TEST1 .BAS	4K	09:34-22 Jan	16:27-30 Jan	09:35-22 Jan	
B1: TEST2 .BAS	4K	11:55-01 Feb		11:55-01 Feb	

When did we print that letter?

Has the mailing list been updated?

Which is the latest version?

DateStamper™ keeps your CP/M computer up-to-date!

- avoid erasing the wrong file
- keep dated tax log of computer use
- back-up files by date and time
- simplify disk housekeeping chores

OPERATION: DateStamper extends CP/M 2.2 to automatically record date and time a file is created, read or modified. DateStamper reads the exact time from the real-time clock, if you have one; otherwise, it records the order in which you use files. Disks prepared for datestamping are fully compatible with standard CP/M.

INSTALLATION: Default (relative-clock) mode is automatic. Configurable for any real-time clock, with pre-assembled code supplied for all popular models. Loads automatically at power-on.

UTILITIES: Enhanced SuperDirectory • Powerful, all-function DATSWEEP file-management program with date and time tagging • Installation and configuration utilities

PERFORMANCE: Automatic. Efficient. Versatile. Compatible.

Requires CP/M 2.2. Uses less than 1K memory. Real-time clock is optional.

When ordering please specify format

8" SSSD, Kaypro, Osborne Formats \$49

For other formats (sorry, no 96 TPI) add \$5.

Shipping and handling \$3

California residents add 6% sales tax

MasterCard and Visa accepted

Specialized versions of this and other software available for the Kaypro.
CP/M is a registered trademark of Digital Research, Inc.

Write or call for further information

Plu'Perfect Systems

BOX 1494 • IDYLLWILD, CA 92349 • 714-659-4432

Circle no. 69 on reader service card.


```

214A: A2 02      234      LDX  #2
214C:          235      *
214C:          236      * READ OR WRITE ONE 512 BYTE
214C:          237      * PRODOS BLOCK.
214C:          238      *
214C: 8C D0 C0    239      SSDIO  STY  SSDLOW
214F: 84 4A      240      STY  TEMP
2151:          241      *
2151:          242      * CHECK A-REG TO SEE IF WE NEED
2151:          243      * TO DO A READ OR WRITE.
2151:          244      *
2151:          245      * A-REG = 101 FOR READ
2151:          246      * A-REG = 102 FOR WRITE
2151:          247      *
2151:          248      * ROTATE A-REG THRU CARRY AND
2151:          249      * CHECK IF CARRY SET OR CLEAR.
2151:          250      * IF CARRY SET THEN READ. IF
2151:          251      * CARRY CLEAR THEN WRITE.
2151:          252      *
2151: A5 42      253      LDA  COMMAND
2153: 4A          254      LSR
2154: 90 08      2161  255      BCC  WRITEBYTE
2156:          256      *
2156: A0 00      257      READBYTE LDY  #0
2158: B1 48      258      LDA  (RAMBANK),Y
215A: A4 4A      259      LDY  TEMP
215C: 91 44      260      STA  (BUFFADR),Y
215E: 18        261      CLC
215F: 90 08      2169  262      BCC  CONT
2161:          263      *
2161:          264      * WRITE BYTE TO RAMDISK
2161:          265      *
2161: B1 44      266      WRITEBYTE LDA  (BUFFADR),Y
2163: A0 00      267      LDY  #0
2165: 91 48      268      STA  (RAMBANK),Y
2167: A4 4A      269      LDY  TEMP
2169:          270      *
2169:          271      * COMMON CODE TO INC TO NEXT
2169:          272      * BYTE TO READ OR WRITE
2169:          273      *
2169: C8          274      CONT
216A: D0 E0      214C  275      BNE  SSDIO
216C: E4 45      276      *
216E: E4 47      277      INC  BUFFADR+1
2170: A5 47      278      INC  BLOCKNUM+1
2172: 8D D1 C0    280      LDA  BLOCKNUM+1
2175: CA          281      STA  SSDHI
2177: D0 D4      214C  282      DEX
2178:          283      BNE  SSDIO
2178: 84 4A      284      STY  TEMP
217A:          285      *
217A:          286      * THIS PART OF THE ROUTINE IS
217A:          287      * ENTERED IF THE STATUS, READ
217A:          288      * OR WRITE I/O IS COMPLETE.
217A:          289      *
217A:          290      * SSD EXIT RETURN
217A:          291      *
217A: A4 4A      292      EXIT  LDY  TEMP
217C: A2 00      293      LDX  #0
217E: 68          294      EXI  PLA
217F: 95 42      295      STA  COMMAND,X
2181: E8          296      INX

2182: E0 09      297      CPY  #SAVEBYTES+1
2184: D0 F8      217E  298      BNE  EX1
2186:          299      *
2186: AD D2 C0    300      LDA  #C0D2
2188: 18        301      CLC
218A: 98        302      TYA
218C: F0 01      218E  303      BEQ  EX2
218D: 38        304      SEC
218E:          305      *
218E: 60        306      EX2  RTS
218F: EA        307      ENDS
2190:          308      *
2190:          309      * *****
2190:          310      *
2190:          311      * THIS IS THE MAIN SET OF JSR
2190:          312      * STATEMENTS TO INSTALL AND
2190:          313      * INIT THE RAMDISK.
2190:          314      *
2190: 20 50 22     315      MAIN  JSR  CHECKDEV
2193: 20 97 23     316      JSR  WHERETO
2196: 20 4F 25     317      JSR  TITLE
2199: 20 44 25     318      JSR  DISPVOL
219C:          319      *
219C: 20 B4 21     320      JSR  RELOCATE
219F: 20 C1 21     321      JSR  SETDEV
21A2: 20 D7 21     322      JSR  SETVECTOR
21A5:          323      *
21A5:          324      * CHECK CLEAR DIRECTORY FLAG
21A5:          325      * IF CLEARDIR = 1 THEN SKIP
21A5:          326      * THE ROUTINES TO WRITE A NEW
21A5:          327      * DIRECTORY TO THE RAMDISK AND
21A5:          328      * SKIP THE ROUTINE TO WRITE A
21A5:          329      * NEW BITMAP TO THE RAMDISK.
21A5:          330      *
21A5: AD 4A 24     331      LDA  CLEARDIR
21A8: D0 09      21B3  332      BNE  MNI
21AA:          333      *
21AA:          334      * DO A NORMAL FIRST TIME INIT
21AA:          335      *
21AA: 20 E2 21     336      JSR  CLEAREND
21AD: 20 EF 21     337      JSR  WRITEDIR
21B0: 20 35 22     338      JSR  BITMAP
21B3: 60          339      RTS
21B4:          340      *
21B4:          341      * *****
21B4:          342      *
21B4:          343      * THIS ROUTINE IS USED TO MOVE
21B4:          344      * THE SSD RAM DISK DRIVER TO
21B4:          345      * THE ADDRESS OF SSDCODE.
21B4:          346      * THE ROUTINE IS COMPLETELY
21B4:          347      * RELOCATABLE AND DOES NOT HAVE
21B4:          348      * TO BE ON A PAGE BOUNDARY.
21B4:          349      *
21B4: A0 00      350      RELOCATE LDY  #0
21B6: B9 00 21     351      LDA  SSDSTART,Y
21B9: 91 02      352      STA  (SSDCODE),Y
21BB: C8          353      INY
21BC: C0 8F      354      CPY  #ENDSSD-SSDSTART
21BE: D0 F6      21B6  355      BNE  R1
21C0: 60          356      RTS
21C1:          357      *

```

(Continued on next page)

WINDOWS FOR DATA™

Featuring One-Step Data Entry♦

Now you can code fast, powerful data entry windows, improve user convenience - reduce input errors.

All the power, convenience and flexibility of the #1 window utility for the IBM PC. Our WINDOWS FOR C™ combined with a professional window-based data entry system.

Complete control over screen display and entry of data within a convenient flexible window environment.

WINDOWS FOR C WINDOWS FOR DATA
(Includes WINDOWS FOR C)

PCDOS	\$ 195	\$ 295
PC/XENIX	\$ 395	\$ 595
UNIX	CALL	CALL

WINDOWS FOR DATA™ provides versatile, easy-to-use data entry functions that operate within windows.

CAPABILITIES INCLUDE:

- Pop-up data entry windows
- Multiple field types
- Data validation functions
- Field-specific & context-sensitive help
- Lotus-style menu design
- Single field entry option
- Date, time and string utilities
- Dynamic control of data-entry environment

- ♦ User input to data-structure variables without intervening code.



**Vermont
Creative
Software**

21 Elm Ave.
Richford, VT 05476
802-848-7738, ext. 31

Full source available. Master Card & Visa accepted. Shipping \$ 3.50. VT residents add 4% tax.

Ram Disk Listing

(Listing continued, text begins on page 84)

```

21C1: 358 *****
21C1: 359 *
21C1: 360 * THIS ROUTINE IS USED TO MAKE
21C1: 361 * ROOM IN THE DEVICE LIST OF
21C1: 362 * ACTIVE DEVICES FOR THE RAM
21C1: 363 * DISK DRIVE IDENTIFICATION.
21C1: 364 *
21C1: 365 SETDEV LDX DEVCNT
21C1: 366 INX
21C1: 367 STX DEVCNT
21C1: 368 *
21C1: 369 SDI LDA DEVLST-1,X
21C1: 370 STA DEVLST,X
21C1: 371 DEX
21C1: 372 BNE SDI
21D1: 373 *
21D1: 374 * STORE THE RAM DISK IN LIST
21D1: 375 *
21D1: 376 LDA #150
21D1: 377 STA DEVLST
21D1: 378 RTS
21D1: 379 *
21D1: 380 *****
21D1: 381 *
21D1: 382 * INSTALL RAMDISK DRIVER VECTOR
21D1: 383 *
21D1: 384 SETVECTOR LDA SSDCODE
21D1: 385 STA DEVADR51
21D1: 386 LDA SSDCODE+1
21D1: 387 STA DEVADR51+1
21D1: 388 RTS
21E1: 389 *
21E1: 390 *****
21E1: 391 *
21E1: 392 * CLEAR BUFFER FROM END OF DISK
21E1: 393 * DATA TO END OF DISK BUFFER.
21E1: 394 *
21E1: 395 CLEAREND LDY #0
21E1: 396 TYA
21E1: 397 CE STA MIDBUFF,Y
21E1: 398 STA MIDBUFF+$100,Y
21E1: 399 INY
21E1: 400 BNE CE
21E1: 401 RTS
21E1: 402 *
21E1: 403 *****
21E1: 404 *
21E1: 405 * CALL MLI TO WRITE OUT THE
21E1: 406 * SSD RAM DISK VOLUME DIRECTORY
21E1: 407 *
21E1: 408 WRITEDIR LDA #DIRBLOCKS

```

```

21F1: 8D 7E 26 409 STA BUFFER+39
21F1: A9 02 410 LDA #502
21F1: 8D 4F 26 411 STA WRITEBLOCK
21F1: 20 00 BF 412 *
21F1: 81 413 JSR MLI
21F1: 4B 26 414 DB #81
21F1: 415 DW MLIWRITE
21F1: 416 *
21F1: 417 * CLEAR AND SET BLOCKS
21F1: 418 * J THRU DIRBLOCKS OF
21F1: 419 * VOLUME DIRECTORY. WE HAVE TO
21F1: 420 * SET UP THE LAST/NEXT BLOCK
21F1: 421 * NUMBERS IN THE FIRST FOUR (4)
21F1: 422 * BYTES OF EACH BLOCK.
21F1: 423 *
21F1: A2 03 424 LDX #3
21F1: 20 28 22 425 CB1 JSR CLEARBUFF
21F1: 2204: 426 *
21F1: 2204: CA 427 DEX
21F1: 2205: 8E 57 26 428 STX BUFFER
21F1: 2208: E8 429 INX
21F1: 2209: E8 430 INX
21F1: 220A: 431 *
21F1: 220A: E0 06 432 CPX #DIRBLOCKS
21F1: 220C: D0 07 2215 433 BNE CB2
21F1: 220E: 434 *
21F1: 220E: A9 00 435 LDA #0
21F1: 2210: 8D 59 26 436 STA BUFFER+2
21F1: 2213: F0 03 2218 437 BEQ CB3
21F1: 2215: 438 *
21F1: 2215: 8E 59 26 439 CB2 STX BUFFER+2
21F1: 2218: 440 *
21F1: 2218: 441 * WRITE VOLUME DIRECTORY BLOCK
21F1: 2218: 442 * TO SSD RAM DISK.
21F1: 2218: 443 *
21F1: 2218: CA 444 CB3 DEX
21F1: 2219: 8E 4F 26 445 STX WRITEBLOCK
21F1: 221C: 446 *
21F1: 221C: 20 00 BF 447 JSR MLI
21F1: 221F: 81 448 DB #81
21F1: 2220: 4B 26 449 DW MLIWRITE
21F1: 2222: 450 *
21F1: 2222: E8 451 INX
21F1: 2223: E0 06 452 CPX #DIRBLOCKS
21F1: 2225: D0 DA 2201 453 BNE CB1
21F1: 2227: 60 454 RTS
21F1: 2228: 455 *
21F1: 2228: 456 *****
21F1: 2228: 457 *
21F1: 2228: 458 * CLEAR ENTIRE FILE BUFFER
21F1: 2228: 459 *

```

ATTENTION

C-PROGRAMMERS

File System Utility Libraries

All products are written entirely in K& RC. Source code included, No Royalties, Powerful & Portable.

BTree Library

75.00

- High speed random and sequential access.
- Multiple keys per data file with up to 16 million records per file.
- Duplicate keys, variable length data records.

ISAM Driver

40.00

- Greatly speeds application development.
- Combines ease of use of database manager with flexibility of programming language.
- Supports multi key files and dynamic index definition.
- Very easy to use.

Make

59.00

- Patterned after the UNIX utility.
- Works for programs written in every language.
- Full macros, File name expansion and built in rules.

Full Documentation and Example Programs Included.

For more information call or write:

softfocus

Credit cards accepted.

1343 Stanbury Drive
Oakville, Ontario, Canada
L6L-2J5
(416) 825-0903
(416) 844-2610

Dealer inquiries invited.

Introducing SMK™, the

SEIDL MAKE UTILITY

SMK is an automatic software product generation utility for MS™-DOS. With SMK, you can update and track every module in your program. When changes are made to any module, SMK will recompile or reassemble changed modules and modules effected by the changes, and relink the modules in the proper order. SMK will work with any compiler, assembler and linker.

Advanced SMK Features Include:

- Proprietary dependency algorithm analyzes all dependencies before rebuilding any files.
- SMK understands complicated dependencies involving nested include files, source, and object code libraries.
- High-level dependency definition language makes setting up dependencies easy. Supports parameterized macros, local variables, constants, include files, command line parameters, line and block comments.
- Batch source code editor that allows automatic source file updating is included.
- Full pathname and directory support.
- Typeset user's manual and excellent error diagnostics make SMK easy to learn and easy to use.

**SMK Introductory Price —
Save 40% — \$84.00**

SMK List Price (effective Jan. 86): \$140.00
(include \$3.50 postage & handling)

SEIDL COMPUTER ENGINEERING

1163 E. Ogden Ave., Suite 705-171
Naperville, IL 60540
(312) 983-5477


```

2228:A0 00      460 CLEARBUFF LDY  #0
2229:98         461 TYP
222B:99 57 24   462 CL1 STA BUFFER,Y
222E:99 57 27   463 STA BUFFER+4100,Y
2231:C8         464 INY
2232:D0 F7      465 BNE CL1
2234:60         466 RTS
2235:          467 *
2235:          468 *****
2235:          469 *
2235:          470 * SET ALL BLOCKS NOT USED IN
2235:          471 * RAM DISK TO HEX $FF TO SHOW
2235:          472 * THAT THOSE BLOCKS ARE FREE.
2235:          473 * SET FIRST BYTE TO SHOW THAT
2235:          474 * BLOCKS 0 THRU DIRBLOCKS ARE
2235:          475 * IN USE.
2235:          476 *
2235:A9 01      477 BITMAP LDA #401
2237:8D 57 24   478 STA BUFFER
223A:A0 3F      479 LDY #43
223C:A9 FF      480 LDA #FF
223E:99 57 26   481 *
2241:88         482 BM2 STA BUFFER,Y
2242:D0 FA      483 DEY
2244:          484 BNE BM2
2244:          485 *
2244:A9 06      486 LDA #DIRBLOCKS
2246:8D 4F 24   487 STA WRITEBLOCK
2249:          488 *
2249:          489 * WRITE OUT BIT MAP TO SSD
2249:          490 * RAM DISK.
2249:          491 *
2249:20 00 BF   492 JSR MLI
224C:81         493 DB #81
224D:4B 24      494 DB MLIWRITE
224F:          495 *
224F:60         496 RTS
2250:          497 *
2250:          498 *****
2250:          499 *
2250:          500 * THIS ROUTINE IS USED TO CHECK
2250:          501 * FOR A DEVICE DRIVER ALREADY
2250:          502 * INSTALLED FOR SLOT AND DRIVE
2250:          503 * WE WANT TO USE. I USE SLOT 5
2250:          504 * DRIVE 1 FOR THE RAMDISK DRIVER
2250:          505 * IF THE ADDRESS AT SLOT 5 DRIVE
2250:          506 * 1 DOES NOT POINT TO ID0A2 THE
2250:          507 * PROGRAM WILL ABORT WITH AN
2250:          508 * ERROR MESSAGE. IF WE FIND NO
2250:          509 * DRIVER INSTALL I READ BLOCK 2
2250:          510 * OF THE RAMDISK AND COMPARE IT
2250:          511 * WITH THE VOLUME DIRECTORY
2250:          512 * HEADER INFORMATION AT BUFFER.
2250:          513 * IF THE INFORMATION IS THE SAME
2250:          514 * A MESSAGE IS PRINTED ASKING
2250:          515 * THE USER IF HE WANTS TO ZERO
2250:          516 * THE DIRECTORY ON THE RAMDISK
2250:          517 * OR NOT. IF WE WANT TO CLEAR
2250:          518 * THE DIRECTORY I SET CLEARDIR
2250:          519 * TO ZERO. IF WE DON'T WANT TO

```

```

2250:          520 * CLEAR THE DIRECTORY I SET
2250:          521 * CLEARDIR TO ONE.
2250:          522 *
2250:          523 CHECKDEV LDA DEVA5R51
2253:C9 A2      524 CMP #NODEV
2255:D0 7A      525 BNE DEVA5R51
2257:          526 *
2257:AD 1B BF   527 LDA DEVA5R51+1
225A:C9 D0      528 CMP #NODEV
225C:D0 73      529 BNE DEVA5R51
225E:          530 *
225E:          531 * LETS TRY AND READ THE RAM DISK
225E:          532 * DIRECTORY AT BLOCK 2 AND
225E:          533 * COMP IT TO SOME OF THE BYTES
225E:          534 * AT ADDR BUFFER TO SEE IF THE
225E:          535 * RAMDISK MIGHT HAVE A FORMATTED
225E:          536 * DIRECTORY ON IT.
225E:          537 *
225E:A9 01      538 LDA #READ
2260:85 42      539 STA #COMMAND
2262:          540 *
2262:          541 * ADDRESS OF READ BUFFER
2262:          542 *
2262:A9 82      543 LDA #MIDBUFF
2264:85 44      544 STA BUFFADR
2266:A9 26      545 LDA #MIDBUFF
2268:85 45      546 STA BUFFADR+1
226A:          547 *
226A:          548 * BLOCK NUMBER TO READ
226A:          549 *
226A:A9 02      550 LDA #02
226C:85 46      551 STA BLOCKNUM
226E:A9 00      552 LDA #00
2270:85 47      553 STA BLOCKNUM+1
2272:          554 *
2272:          555 * READ THE RAMDISK USING DRIVER
2272:          556 * AT SSDSTART IN THIS PROGRAM
2272:          557 *
2272:20 00 21   558 JSR SSDSTART
2275:          559 *
2275:          560 * WE WILL NOW ASSUME AT THERE
2275:          561 * IS THE INFORMATION IN BLOCK 2
2275:          562 * OF THE RAMDISK AT ADDRESS
2275:          563 * MIDBUFF. COMPARE THREE SETS OF
2275:          564 * BYTES TO SEE IF IT MATCHES A
2275:          565 * VOLUMN DIRECTORY HEADER BLOCK.
2275:          566 * IF ANY BYTE DOES NOT MATCH
2275:          567 * THE ROUTINE JUMPS TO NODIR
2275:          568 * IF ALL THREE SETS OF BYTES
2275:          569 * MATCH I ASK USER IF HE WANTS
2275:          570 * TO CLEAR THE DIR OR NOT.
2275:          571 *
2275:A2 00      572 LDY #00
2277:BD 57 26   573 CD0 LDA BUFFER,X
227A:DD 82 26   574 CMP MIDBUFF,X
227D:D0 4C      575 BNE NODIR

```

(Continued on next page)

Sidekick for CP/M! Write-Hand-Man

Desk Accessories for CP/M
NEW! Now with automatic screen refresh!

Suspend CP/M applications such as WordStar, dBase, and SuperCalc, with a single keystroke and look up phone numbers, edit a notepad, make appointments, view files and directories, communicate with other computers, and do simple arithmetic. Return to undisturbed application! All made possible by Write-Hand-Man. Ready to run after a simple terminal configuration! No installation required.

Don't be put down by 16 bit computer owners. Now any CP/M 2.2 machine can have the power of Sidekick.

Bonus! User extendable! Add your own applications.

\$49.95 plus tax (California residents), shipping included!
Volume and dealer discounts.

Available on IBM 8 inch and Northstar 5 inch disks. Other 5 inch formats available with a \$5.00 handling charge. CP/M 2.2 required; CP/M 3 not supported.

COD or checks ok, no credit cards

Poor Person Software

3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735

Write-Hand-Man trademark of Poor Person Software, CP/M trademark of Digital Research, Sidekick trademark of Borland International, dBase trademark of Ashton-Tate, WordStar trademark of Micropro, SuperCalc a trademark of Sorcim.

Wizard C

Discover the powers of Wizard C
for yourself!

"...written by someone who has been in the business a while. This especially shows in the documentation."

Computer Language
February, 1985

" Wizard's got the highest marks for support."

"The Wizard compiler had excellent diagnostics; it would be easier writing portable code with it than with any other compiler we tested."

Dr. Dobb's Journal
August, 1985

Full Lint checking, six memory models, 8087 support, in-line assembly language, ROMable data support, full library source code. Cross-compilers are available on VAX/VMS and UNIX machines.

(617) 641-2379

Only \$450.

WIZARD
SYSTEMS SOFTWARE, INC.

11 Willow Court
Arlington, MA 02174




```

227F: E8 576 *
2280: E0 04 577 INX
2282: D0 F3 2277 578 CPX #004
2284: 580 * BNE CD0
2284: A2 14 581 LDX #014
2286: BD 57 26 582 CD1 LDA BUFFER,X
2289: DD 82 26 583 CMP MIDBUFF,X
228C: D0 3D 22CB 584 BNE NODIR
228E: 585 *
228E: E8 586
228F: E0 25 587 INX
2291: D0 F3 2284 588 CPX #025
2293: A2 27 589 BNE CD1
2295: BD 57 26 590 LDX #027
2298: DD 82 26 591 CD2 LDA BUFFER,X
229B: D0 2E 22CB 592 CMP MIDBUFF,X
229D: 593 BNE NODIR
229D: E8 594
229E: E0 2B 595 INX
22A0: D0 F3 2295 596 CPX #02B
22A2: 597 BNE CD2
22A2: 598 *
22A2: 599 * ASK USER IF HE WANT TO ZERO
22A2: 600 * DIRECTORY OR NOT IF THE USER
22A2: 601 * WANTS TO ZERO THE DIRECTORY
22A2: 602 * JUST CONTINUE WITH THE INSTALL
22A2: 603 * PROGRAM. IF THE USER DOES NOT
22A2: 604 * WANT TO ZERO THE DIRECTORY
22A2: 605 * DO A JUMP TO THE DOS WARMSTART
22A2: 606 *
22A2: 20 58 FC 607 JSR HOME
22A5: 20 3A FF 608 JSR BELL
22A8: 609 *
22A8: A0 00 610 LDY #0
22AA: B9 EF 22 611 CD3 LDA DIRFOUND,Y
22AD: C9 00 612 CMP #EOL
22AF: F0 09 22BA 613 BEQ CD4
22B1: 614 *
22B1: 09 80 615 ORA #180
22B3: 20 ED FD 616 JSR COUNT
22B4: C8 617 INY
22B7: 4C AA 22 618 JMP CD3
22BA: 619 *
22BA: 20 41 25 620 CD4 JSR GETKEY
22BD: C9 59 621 CMP #Y
22BF: F0 0A 22CB 622 BEQ NODIR
22C1: 623 *
22C1: C9 4E 624 CMP #N
22C3: D0 F5 22BA 625 BNE CD4
22C5: 626 *
22C5: 627 * THE USER DOES NOT WANT TO
22C5: 628 * CLEAR THE DIRECTORY.
22C5: 629 *
22C5: A9 01 630 LDA #01

```

```

22C7: 8D 4A 26 631 STA CLEARDIR
22CA: 60 632 RTS
22CB: 633 *
22CB: 634 * WE WANT TO CLEAR THE DIRECTORY
22CB: 635 * ON THE RAMDISK SO SET FLAG FOR
22CB: 636 * CLEAR AND RETURN TO ROUTINE.
22CB: 637 *
22CB: A9 00 638 NODIR LDA #00
22CD: 8D 4A 26 639 STA CLEARDIR
22D0: 60 640 RTS
22D1: 641 *
22D1: 642 * THIS PART OF THE ROUTINE IS
22D1: 643 * ENTERED IF WE FIND A DEVICE
22D1: 644 * DRIVER ADDRESS THAT DOES NOT
22D1: 645 * POINT TO THE NO DEVICE DRIVER
22D1: 646 * ACTIVE ADDRESS OR TO THE
22D1: 647 * ADDRESS WHERE WE WANT TO PUT
22D1: 648 * OUR CODE. THERE COULD BE A
22D1: 649 * HARD DISK OR OTHER STORAGE
22D1: 650 * DEVICE ALREADY INSTALL FOR
22D1: 651 * THAT SLOT/DRIVE.
22D1: 652 *
22D1: 20 58 FC 653 DEVBABORT JSR HOME
22D4: 20 3A FF 654 JSR BELL
22D7: 655 *
22D7: A0 00 656 LDY #0
22D9: B9 4B 23 657 DA1 LDA ABORTMSG,Y
22DC: C9 00 658 CMP #EOL
22DE: F0 09 22E9 659 BEQ DA2
22E0: 660 *
22E0: 09 80 661 ORA #180
22E2: 20 ED FD 662 JSR COUNT
22E5: C8 663 INY
22E6: 4C D9 22 664 JMP DA1
22E9: 665 *
22E9: 20 3A FF 666 DA2 JSR BELL
22EC: 4C 00 BE 667 JMP WARMDS
22EF: 668 *
22EF: 669 *****
22EF: 670 *
22EF: 671 * DIRECTORY FOUND MESSAGE
22EF: 672 *
22EF: 8D 8D 8D 8D 673 DIRFOUND DFB CR,CR,CR,CR
22F3: 8D 8D 8D 8D 674 DFB CR,CR,CR,CR
22F7: 20 20 20 20 675 ASC '
2303: 44 49 52 45 676 ASC 'DIRECTORY FOUND'
2312: 8D 8D 677 DFB CR,CR
2314: 20 20 20 20 678 ASC '

```

(Continued on page 92)

SUPER-CHARGE your directories for only \$39.95

Say goodbye to path names and unleash the full power of DOS on your IBM Personal Computer. With SuperPATH™ from Martin Scot, you can run any program from any directory, from any disk, anytime.

SEE THE BIG PICTURE.

Tired of playing hide and seek with your files? SuperPATH tells you where your files are, instantly. Anywhere on your disk.

EXPLOSIVE PERFORMANCE.

Program your directories. To display just the files you want to see, in any format, sorted or double sorted. To set up your keyboard macros for each new directory you enter. To back up your files automatically when you leave a directory.

Or, you can use SuperPATH's batch building option to create your own file management utilities.

NO RISK.

Comes with a 30 day money-back guarantee.

CALL (206) 527-9605 to order SuperPATH.

MARTIN SCOT

4515 Purdue N.E. Seattle, WA 98105

RP/M2™ creates

CP/M©2.2 compatible

IBM PC

1. Remove the 8088
2. Install the NEC μ PD70108
3. Boot PC RP/M2

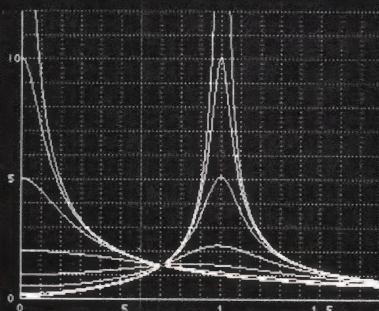
MSDOS and CP/M 2.2 capability coexist in your IBM PC. The NEC μ PD70108 CPU chip is a fast 8088 that can also execute 8080 machine code. PC RP/M2 is standard 2.2 compatible RP/M2 with our CBIOS for the IBM PC. System disk with manual and NEC μ PD70108 \$129. Shipping \$5 (\$10 nonUS)

**Micro
Methods, Inc.**

118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

isys forth

for the Apple®][



Parallel Resonance with Damping
BASIC 213 sec ISYS FORTH 20 sec



Hydrogen 3p Orbital Cross-section
BASIC 492 sec ISYS FORTH 39 sec

ISYS FORTH is a FORTH-83 compiler designed especially for scientific and engineering applications.

- **FASTEST** — Sieve benchmark, 3.3 seconds/pass. Compiles to machine language.
- **FLOATING POINT WITH TRANSCENDENTALS**, single and double precision.
- **16-BIT 65802 SUPPORT** for execution speeds 50-100% faster than the above.
- **FIXED POINT THEORY AND EXAMPLES** (a 30-page chapter). D*! for double precision fixed point.
- **TURTLE AND CARTESIAN GRAPHICS** with 70-column character set and double hires support.
- **MACRO ASSEMBLER**.
- **DOS 3.3 FILE INTERFACE**.
- **FULL-SCREEN EDITOR**.
- **144-PAGE MANUAL WITH TUTORIAL**.
- **PRICE: \$99, NO EXTRA CHARGES.**

ILLYES SYSTEMS

PO Box 2516, Sta A
Champaign, IL 61820
Phone: 217/359-6039

For any Apple][model, 48K or larger. Apple is a registered trademark of Apple Computer.

Circle no. 99 on reader service card.

We will do whatever it takes to make DSD86 the best debugger available for the IBM PC.

For starters, we have by far the best design, a superior base to build from.

While the competition adds new "modes" for every feature, we have a pure, consistent and expandable design. While the competition forces you to accept their particular philosophy, we offer maximized flexibility. If you already have a debugger or are looking for your first, look no further because you can't do any better. We invite you to compare our debugger, DSD86, with any other on the market.

- Recursive Command Macros & Files ■
- Bind Macros to any key ■
- Multi-segment Symbol Support ■
- Symbolic Register & Stack Displays ■
- User Customizable Screen Layout ■
- Superior Mode-less Design ■
- Source Window for MS Languages ■
- User Writable Commands & Displays ■
- Fast Screen Update ■
- Unique Breakpointing Facilities ■
- 30 Day Money Back Guarantee ■

Call or write for our free report on truly advanced debugging technology which explains DSD86's design and why it is superior to the debugger you are currently using.

Take the DSD challenge: secure a money back guarantee with any of our competitors. Buy both debuggers and use them for a month. Send the one you like least back for a refund.

Only \$69.95!

Soft Advances
P.O. Box 49473
Austin, Texas 78765
512-478-4763

"Programming for Productivity and Profit"

Please include \$4 shipping

Circle no. 83 on reader service card.

ATTENTION!

DDT, SID and ZSID USERS!

Why are you wasting your time with outdated debugging tools when you could be using DSD80 by Soft Advances?

DSD80 is unmatched in the CP/M-80 world and even makes most 16 bit PC programmers green with envy.

- Full Screen Display ■
- DDT Command Compatibility ■
- Software In-Circuit Emulator ■
- Superior Symbol Support ■
- Intel and Zilog Mnemonics ■
- 8080 and Z80 Support ■
- Single Step Subroutines ■
- ASCII and Binary String Searching ■
- Port Input and Output ■
- View Disk Files ■
- Uses only 16k of TPA ■
- Stack Display ■
- Highlighted Memory Display ■
- Dual Monitor Support ■
- User Writable Display ■
- 30 Day Money Back Guarantee ■

You can experience the power of DSD80 for only \$125. Our money back guarantee will protect you from overloads.

Soft Advances
P.O. Box 49473
Austin, Texas 78765
512-478-4763

"Programming for Productivity and Profit"

Please include \$4 for shipping. DSD is a trademark of Soft Advances; CP/M, SID & ZSID are trademarks of Digital Research. Dealer inquiries invited. Ask us about DSD86 for the IBM PC.

Circle no. 111 on reader service card.


```

231A 43 4C 45 41 679      ASC 'CLEAR      DIRECTORY "Y" OR "N"
2334 8D 8D 8D 8D 680      DFB CR,CR
2336 20 20 20 20 681      ASC '
234A 00 682      DFB EOL
234B: 683 *
234B: 684 * ABORT MESSAGE
234B: 685 *
234B: 8D 8D 8D 8D 686 ABORTMSG DFB CR,CR,CR,CR
234F 8D 8D 8D 8D 687 DFB CR,CR,CR,CR
2353 20 20 20 20 688      ASC '
2357 44 45 56 49 689      ASC 'DEVICE      DRIVER ALREADY INSTALLED'
2376 8D 8D 690      DFB CR,CR
2378 20 20 20 20 691      ASC '
2385 50 52 4F 47 692      ASC 'PROGRAM      ABORT'
2392 8D 8D 8D 8D 693 DFB CR,CR,CR,CR
2396 00 694      DFB EOL
2397: 695 *
2397: 696 *****
2397: 697 *
2397: 698 * THIS ROUTINE IS USED TO ASK
2397: 699 * THE USER WHERE HE/SHE WOULD
2397: 700 * LIKE TO MOVE THE RAMDISK
2397: 701 * DRIVER CODE TO THERE ARE
2397: 702 * THREE LOCATIONS TO MOVE THE
2397: 703 * CODE TO OR THE USER CAN QUIT
2397: 704 *
2397: 705 * LOC 1: INSIDE PRODOS AT $F000
2397: 706 * LOC 2: BETWEEN THE BASIC
2397: 707 * INTERPRETER AND THE
2397: 708 * FILE BUFFERS
2397: 709 * LOC 3: AT $300
2397: 710 *
2397: 20 58 FC 711 WHERETO: JSR HOME
239A A0 00 712      LDY #0
239C B9 0F 24 713 WT0      LDA WHERMSG,Y
239F C9 00 714      CMP #EOL
23A1 F0 09 23AC 715      BEQ WT1
23A3: 716 *
23A3: 09 80 717      ORA #580
23A5 20 ED FD 718      JSR COUT
23A8 C8 719      INY
23A9 4C 9C 23 720      JMP WT0
23AC: 721 *
23AC: 20 3A FF 722 WT1      JSR BELL
23AF 20 41 25 723      JSR GETKEY
23B2 C9 31 724      CMP #'1'
23B4 F0 15 23CB 725      BEQ WT2
23B6: 726 *
23B6: C9 32 727      CMP #'2'
23B8 F0 20 23DA 728      BEQ WT3
23BA: 729 *
23BA: C9 33 730      CMP #'3'
23BC F0 2A 23E8 731      BEQ WT4
23BE: 732 *

```

(Continued on page 94)

¿C? ¡SÍ!

If you're a C programmer (or want to be one), we speak your language. Subscribe to **The C Journal** today, and start increasing your productivity right away. We give you information you can use on any machine — IBM PC™, UNIX™-based, Macintosh™, or CP/M™ — micro, mini, or mainframe.

- in-depth reviews and feature articles — C compilers, editors, interpreters, function libraries, and books.
- hints and tips — help you work **better** and **faster**.
- interviews — with software entrepreneurs that **made it** — by using C.
- news and rumors — from the ANSI standards committee and the industry.

Limited Time Offer

Join our thousands of subscribers at the **Discount Rate** of only \$18 for a full year (regularly \$28)! Call us now at (201) 989-0570 for faster service — don't miss a single issue of **The C Journal**!

Please add \$9 for overseas airmail.

Trademarks — CP/M: Digital Research Inc. IBM PC: IBM Corp. Macintosh: Apple Computer Corp. **The C Journal**: InfoPro Systems. UNIX: AT&T Bell Labs.



InfoPro Systems
3108 Route 10
Denville, NJ 07834
(201) 989-0570



Decmation's

BLUE LIGHTNIN'

PC Z80 CO-PROCESSOR

CP/M* EMULATOR

The *Blue Lightnin'* blows the competition away with its 9 MHz clock speed, 128K byte memory and optimized I/O.

With the *Blue Lightnin'* your PC or AT compatible can run CP/M-80 software, or optionally, Intel development system software (ISIS). We include a debugger and a quality Z80 assembler and linker. You also get an Xmodem-compatible serial communications program, in both CP/M and MS-DOS versions. A version for Intel development systems is also available.

We have plug-in cards for VAX** and PDP-11** to run CP/M-80 or MS-DOS software.

The ideal tool for microprocessor software development or any CP/M application.

The *Blue Lightnin'* is distributed by:

TSI 1897 Garden Ave., Eugene, OR 97403
1-800-874-2288

The *Blue Lightnin'* is manufactured by:

(408) 980-1678

Decmation



Telex 171596
AAA COM SUVL

2065 Martin Avenue, # 110 Santa Clara, California 95050

*CP/M is a registered trademark of Digital Research, Inc.
**VAX and PDP-11 are trademarks of the Digital Equipment Corporation, with whom Decmation is not connected.

Circle no. 75 on reader service card.

Circle no. 162 on reader service card.



MacTutor

The Macintosh Programming Journal

"...the anticipation of finally being able to associate myself with a first rate tech journal leaves me almost 'codeless!'" -R. Onzo

"Ah! Eureka! This is the publication that I've been looking for. It's not too glossy, slick or condescending. It's meat, fact, it's wonderful!" - Rick Barrett

The number one technical journal for Macintosh software development in C, assembly, Pascal, Basic, Forth, with NEON, Modula-2, APL and Lisp. \$24 per year US, \$30 for Canada and \$36 overseas. Find out why we're number 1 today!

Sample: (714) 579-7700

P.O. Box 846, Placentia, CA. 92670

Circle no. 68 on reader service card.

RUN CP/M 80 ON IBM FAST!

Now your IBM PC or compatible can access the vast library of CPM/80-2.2 software titles, and run them at lightning speed! Our revolutionary MICRUN/CPM non-emulating interface executes CP/M software on PC's by utilizing the power of NEC's newest microprocessor, the V-20. The V-20/30 chip replaces your PC resident 8088 or 8086 microprocessor chip, thereby providing 8 bit processing ability and actually improving it's 16 bit performance by up to 50%. If your PC is not already equipped with the NEC V-20 there is no problem. Simply purchase our MICRUN/CPM software package, and we will provide you with the V-20 chip for only \$45 dollars.

Features

- * Incredibly fast-up to 8 MHz
- * Ability to run CPM programs in color
- * Cross sub-directory operations
- * No expensive co-processor board req.
- * Not an emulator
- * Includes disk transfer utility for reading, writing, and formatting up to 70 different CP/M and MS-DOS disks.
- * 16 logical/physical drives
- * Includes RS232 communications program for transferring software to or from CPM system to PC.

Supports

Osborne* Kaypro* Zenith H-19* Televideo* Radio Shack* ADM3A* VT-52* Visual 210 adds Viewpoint* Hazeltine Espirit* 1510* Plus many more

ONLY \$99.95

Included in the above price: Interface software, disk transfer utility program, RS232 communications program, complete documentation. Not included is the NEC V-20 microprocessor chip which is available at a price of only \$45 dollars.



1-800-637-7226



Orders only.



Dealer Inquiries Invited

Trademarks: CPM (Digital Research, Inc.), IBM (IBM Corp.).

Micro Interfaces Corporation
6824 N.W. 169th Street
Hialeah, Florida 33015
(305) 823-8088

Circle no. 110 on reader service card.

ST-FORTH \$49

For your IBM PC/XT/AT/PCjr

A complete FORTH development system for beginners or experienced users

- 100% FORTH-83 Standard
- ALL source code provided
Written entirely in FORTH & FORTH assembler
- Powerful, fast, easy-to-use editor
Search, locate source, edit errors
- Extensive MS-DOS/PC-DOS interface
Sequential & random files
- Many more useful utilities
- Supplied as MS-DOS .COM file
Compatible with all versions of DOS
- 148-page User's Manual
- Unconditional money-back guarantee

Beginner's pkg based on Leo Brodie's *Starting FORTH* available

Call or write for brochure and order form.



**Sunset
Technology**

1954 Menalto Ave.
Menlo Park, CA 94025
(415) 325-3680

Circle no. 144 on reader service card.

d/MULTI MULTIUSER dBASE for TurboDOS

TRUE File and Record Locking as easy as d-BASE-II. Unlimited users can perform the magic of dBASE in the program or interactive mode

- * TurboDOS 1.3 or 1.4
- * No Peeks or Pokes
- * System Date and Time Functions
- * Printspooler Controls up to 16 printers

Martian Technologies . . .

**.CREATEing Multi-users from
Single-users around the world**

CALL FOR DETAILS

Martian Technologies
8348 Center Dr., Ste.-F, La Mesa, CA 92041
(619) 464-2924



Circle no. 70 on reader service card.


```

23BE:C9 51      233      CMP    #'Q'
23C0:D0 EA      234      BNE    WT1
23C2:          235      *
23C2:20 58 FC      236      JSR    HOME
23C3:20 3A FF      237      JSR    BELL
23C8:4C 00 BE      238      JMP    WARMDS
23CB:          239      *
23CB:          240      * MOVE CODE INSIDE PRODOS
23CB:          241      *
23CB:A9 00      242 WT2      LDA    #000
23CD:85 02      243      STA    SSDCODE
23CF:A9 FF      244      LDA    #FFF
23D1:85 03      245      STA    SSDCODE+1
23D3:AD 81 C0      246      LDA    #C081
23D6:AD 81 C0      247      LDA    #C081
23D9:60          248      RTS
23DA:          249      *
23DA:          250      * MOVE CODE UNDER BI
23DA:          251      *
23DA:A9 01      252 WT3      LDA    #1
23DC:20 F5 BE      253      JSR    GETBUFF
23DF:B0 10      254      BCS    BUFFERROR
23E1:          255      *
23E1:85 03      256      STA    SSDCODE+1
23E3:A9 00      257      LDA    #100
23E5:85 02      258      STA    SSDCODE
23E7:60          259      RTS
23E8:          260      *
23E8:          261      * MOVE CODE TO #300
23E8:          262      *
23E8:A9 00      263 WT4      LDA    #000
23EA:85 02      264      STA    SSDCODE
23EC:A9 03      265      LDA    #03
23EE:85 03      266      STA    SSDCODE+1
23F0:60          267      RTS
23F1:          268      *
23F1:          269      * ERROR GETTING BUFFER FROM SYS
23F1:          270      *
23F1:20 3A FF      271 BUFFERROR JSR    BELL
23F4:20 58 FC      272      JSR    HOME
23F7:A0 00      273      LDY    #0
23F9:B9 B3 24      274 BE0      LDA    NOBUFF,Y
23FC:C9 00      275      CMP    #EOL
23FE:F0 09      276      BEQ    BE1
2400:          277      *
2400:09 80      278      ORA    #80
2402:20 ED FD      279      JSR    COUT
2405:C8          280      INY
2406:4C F9 23      281      JMP    BE0
2409:          282      *
2409:20 3A FF      283 BE1      JSR    BELL
240C:4C 00 BE      284      JMP    WARMDS
240F:          285      *
240F:          286      *****

```

(Continued on page 96)

The LIBRARIAN Inc.

SOURCE INCLUDED

has taken the drudgery out of generating plots. These libraries contain the most complete collection of plotting subroutines you can buy. Each collection places over 40 routines (in FORTRAN or C language*) at your fingertips. The library makes it easy to:

- plot semi-log and log-log graphs
- draw arcs and circles
- scale charts to exact sizes
- label text in different sizes, directions, and orientations
- design bar charts and pie charts
- and much more

Each package includes source code and a comprehensive manual.

Operating Systems: MS-DOS and PC-DOS

Plotters Supported: Houston Instruments
Hewlett-Packard
HPGL compatibles

We are constantly testing our libraries on different computers and with different compilers including: Microsoft FORTRAN, DR FORTRAN-77, RM FORTRAN, Lahey F77L, CI-C86™, and MWC 86. Specify compiler when ordering.

Price: \$295.00

For More Information: The Librarian, Inc.
10435 Greenbough, #110
Stafford, TX 77477
(713) 499-7662 or 491-2088

*We're developing libraries in BASIC and Pascal, too!

Circle no. 117 on reader service card.

The Best
ProDOS Assembler
In The World

ORCA/M™

Who says so?

Among others, *Peelings* // magazine! When they did a comparative study of Apple // assemblers, here's how they rated them:

ORCA/M	AAA
Merlin	A+
S-C Macro Assembler	A+
The Assembler (Micro Sparc)	A
Lisa 2.5	B+
EdASM	B

ORCA/M 4.0

is also the only assembler to fully support the new 65816 CPU - earning it the endorsement of the Western Design Center, designers of the 65C02, 65802 and 65816!

ORCA/M 4.0 is a complete development environment for assembly language programming for ProDOS. The system includes the Assembler with full macro capabilities, Link Editor, Command Processor, Full Screen Text Editor, Disassembler, many utilities, Macro and Subroutine Libraries, (2, 4 and 8 byte integer mathematics and full graphics libraries, including double hi-res) and 6502, 65C02, 65802 and 65816 support!

\$79.95

Small-C Compiler

All full C language statements are supported, along with most operators and short and long integers. Supports P-code for space efficiency and native code for speed. Complete source code for the compiler is included. (Requires ORCA/M 4.0)

\$39.95

Support Programs

MON+ - A powerful Symbolic Debugger with disassembler, mini-assembler, step and trace, and much more for debugging assembly language programs. **\$29.95**

ORCA O/S Source - Includes the source code for the ORCA/M operating system, subroutine libraries and XREF (global cross reference) utility. **\$39.95**

Floating Point Libraries - Supports IEEE single and double precision floating point formats. Includes transcendental functions and complete source code. **\$39.95**

The Byte Works Inc.
8000 Wagon Mound Dr. NW
Albuquerque, NM 87120
(505) 898-8183

New Mexico residents
add 4.375%
Free shipping in
U.S. and Canada

1010 1101 0110



Circle no. 177 on reader service card.

PRESENTING THE MEGAMAX C COMPILER

NOW AVAILABLE FOR THE MACINTOSH

• IN-LINE ASSEMBLY • ONE PASS COMPIATION • SUPPORT OF DYNAMIC OVERLAYS • FULL ACCESS OF MACINTOSH TOOLBOX ROUTINES • AND MUCH MORE ...

DEVELOPMENT SYSTEM PACKAGE INCLUDES:

• FULL-SCALE IMPLEMENTATION (K&R) C COMPILER • THE STANDARD C LIBRARY • ROM ROUTINES LIBRARY • LINKER • LIBRARIAN AND DOCUMENTATION ...

\$299.95 DEALER AND USER GROUP INQUIRES INVITED
FOR MORE INFORMATION OR TO ORDER CALL OR WRITE:

Megamax, Inc.
BOX 851521, DEPT. V
RICHARDSON, TX
75085-1521
(214) 987-4937

MACINTOSH IS A REGISTERED TRADEMARK OF APPLE COMPUTER INC.

Circle no. 84 on reader service card.

UNIX/XENIX COMPATIBILITY!



C-LIB incorporates over 200 routines to extend the capabilities of C on the IBM PC. Versions are available for the Lattice, Microsoft, and DeSmet (C Ware) C compilers. All libraries are compatible with PC-DOS 2.0+ operating systems.

Featuring:

- UNIX-Compatible "CURSES" Screen/Window Processing Library
- Buffered, Interrupt-driven, Asynchronous Communications Library
- Powerful String Functions
- 8087/Microsoft Floating Point Conversion

C-LIB SOFTWARE AND MANUAL PACKAGE \$195.

Please call or write for additional information.
DEMO DISKETTE AVAILABLE, please include \$5.

VANCE info systems

2818 clay street • san francisco, ca 94115 • (415) 922-6539

IBM & PC-DOS are trademarks of International Business Machines Corp. Lattice is a trademark of Lattice Inc. Xenix & Microsoft C are trademarks of Microsoft Inc. UNIX is a trademark of Bell Labs. C-LIB is a trademark of VANCE info systems.

Circle no. 154 on reader service card.

SERIOUS SOFTWARE FOR SERIOUS DEVELOPERS

FABS PLUS

(B-Tree Structure) Rapid access and maintenance of very large files with millions of records. Extremely fast on unlimited number of keys. Index utility provides protection from corrupted index files. Can be used on files as large as your system can hold.

RETAIL PRICE \$195.00

Network versions of FABS PLUS also available for PC-DOS 3. XX and Novell Netware.

RETAIL PRICE \$395.00

AUTOSORT

Sort/Merge subsystem for very large files. In stand-alone or callable sub-routine, extremely fast. Sorts files as large as the operating system and workspace will allow.

RETAIL PRICE \$150.00

Versions available for FABS and AUTOSORT:
CPM-80, CP/M-86, MP/M-11, MS-DOS, PC-DOS, MICROSOFT BASIC(S), COBOL, FORTRAN, PASCAL, PL/1, CBASIC, CB80, CB86, CBASIC86, LATTICE C

DB-FABS

Data, Screen, Report Manager: Can be used in stand-alone and run-time modes. Creates files and forms for the management and access of data. Report and screen handler to generate listings. B-Tree indexing and high speed sort capabilities.

RETAIL PRICE \$295.00

Version of DB-FABS available for Novell Netware

RETAIL PRICE \$395.00

COMPUTER CONTROL SYSTEMS, INC.
ROUTE 3, BOX 168, LAKE CITY, FL 32055 (904) 752-0912

Circle no. 127 on reader service card.

Ram Disk Listing (Listing continued, text begins on page 84)

```

240F: 787 *
240F: 788 * MESSAGE ASKING USER WHERE
240F: 789 * TO PUT RAMDISK DRIVER CODE.
240F: 790 *
240F: 8D 8D 8D 8D 791 WHEREMSG DFB CR,CR,CR,CR
2413: 20 20 20 20 792 ASC
2418: 52 41 4D 44 793 ASC 'RAMDISK DRIVER ADDRESS
242F: 4F 50 54 49 794 ASC 'OPTIONS'
2436: 8D 8D 8D 795 DFB CR,CR,CR
2439: 20 20 20 20 796 ASC
243F: 5B 31 5D 20 797 ASC '[1]
2459: 8D 8D 798 DFB CR,CR
245B: 20 20 20 20 799 ASC
2461: 5B 32 5D 20 800 ASC '[2]
247C: 8D 8D 801 DFB CR,CR
247E: 20 20 20 20 802 ASC
2484: 5B 33 5D 20 803 ASC '[3]
248F: 8D 8D 804 DFB CR,CR
2491: 20 20 20 20 805 ASC
2497: 5B 51 5D 55 806 ASC '[QUIT'
249D: 8D 8D 807 DFB CR,CR
249F: 20 20 20 20 808 ASC
24A9: 20 20 20 20 809 ASC
24B2: 00 810 DFB EOL
24B3: 811 *
24B3: 812 *****
24B3: 813 *
24B3: 814 * NO DOS BUFFER FOUND UNDER BI
24B3: 815 *
24B3: 8D 8D 8D 8D 816 NOBUFF DFB CR,CR,CR,CR
24B7: 8D 8D 8D 8D 817 DFB CR,CR,CR,CR
24BB: 20 20 20 20 818 ASC
24C9: 46 41 54 41 819 ASC 'FATAL ERROR'
24D4: 8D 8D 820 DFB CR,CR
24D6: 20 20 20 20 821 ASC
24D9: 43 4F 55 4C 822 ASC 'COULD NOT ALLOCATE BUFFER BETWEEN
24FA: 8D 8D 823 DFB CR,CR
24FC: 20 20 20 20 824 ASC
2506: 42 49 20 41 825 ASC 'BI AND FILE BUFFERS'
2519: 8D 8D 826 DFB CR,CR
251B: 20 20 20 20 827 ASC
2520: 46 4C 41 53 828 ASC 'FLASHCARD DRIVER NOT INSTALLED'
253E: 8D 8D 829 DFB CR,CR
2540: 00 830 DFB EOL
2541: 831 *
2541: 832 *****
2541: 833 *
2541: 834 * THIS ROUTINE IS USED TO GET
2541: 835 * A SINGLE KEYPRESS FROM THE
2541: 836 * KEYBOARD AND RETURN IT IN
2541: 837 * THE A-REG WITH THE MSB OFF.
2541: 838 *
2541: 8D 10 C0 839 GETKEY STA STROBE
2544: AD 00 C0 840 LDA KEY
2547: 10 FB 2544 841 BPL GK1
2549: 8D 10 C0 842 STA STROBE
254C: 29 7F 843 AND #67F
254E: 60 844 RTS
254F: 845 *
254F: 846 *****
254F: 847 *
254F: 848 * PRINT PROGRAM TITLE
254F: 849 *
254F: 20 58 FC 850 TITLE JSR HOME
2552: A0 00 851 LDY #0
2554: B9 9B 25 852 TI1 LDA MSG,Y
2557: C9 00 853 CMP #EOL
2559: F0 08 2563 854 BEQ TI2
255B: 855 *
255B: 09 80 856 ORA #80
255D: 20 ED FD 857 JSR COUT
2560: C8 858 INY
2561: D0 F1 2554 859 BNE TI1
2563: 60 860 TI2 RTS
2564: 861 *
2564: 862 *****
2564: 863 *
2564: 864 * THIS ROUTINE WILL DISPLAY THE
2564: 865 * RAM DISK VOLUME NAME IF THE

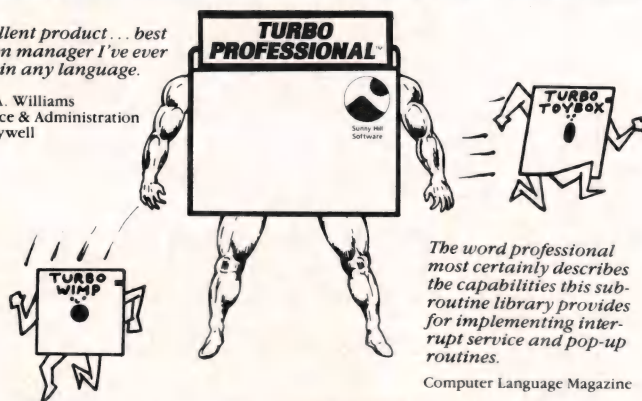
```

(Continued on page 98)

ATTENTION TURBO PASCAL PROGRAMMERS

Excellent product... best screen manager I've ever seen in any language.

Don A. Williams
Finance & Administration
Honeywell



Computer Language Magazine

Get the tools that have no competition.

TURBO PROFESSIONAL's easy to use routines let you... write safe, "sidekickable" routines that can use DOS

- execute DOS commands from Turbo
- create super-fast windowed displays
- service interrupts without assembler
- make your own keyboard enhancers
- allocate memory from DOS
- print concurrently with DOS 3.x.

Complete with 109 + routines, manual, source code to Super Macs keyboard enhancer, and many example programs.

Only \$49.95 + \$5.00 shipping & handling

Visa, MasterCard ok.

Order now!

(206) 367-0650



Sunny Hill Software
13732 Midvale North Suite 206
Seattle, Washington 98133

Requires Turbo for compatibles. Turbo Pascal & Sidekick Trademark Borland Intl.

Circle no. 172 on reader service card.

PC **TEX** Now for PC Users:

Professional Typesetting Capability

PCTeX brings to the personal computer user the ability to put any kind of information on paper in a professional, elegant manner. It brings the full power and flexibility of TeX implementations on mainframes to owners of IBM PC's, AT's and workalike computers. PCTeX is widely used for formatting technical and mathematical material. It is also perfectly suited for producing professional-quality reports, manuals, even books.

PCTeX offers a wide range of typefaces, and a wide choice of drivers which output the finished material on dot matrix printers (Epson, Toshiba), low-cost laser printers (Apple LaserWriter, Corona LP300) and graphics screen preview (Hercules, EGA). All text in this ad has been formatted by PCTeX and produced on a Corona Laser Printer.

Join hundreds of satisfied PCTeX users. Write or call us today.

PCTeX: only \$279. Dot-matrix drivers: \$100. Laser drivers: \$300. Preview (Hercules GC): \$250. MF Medley (44 fonts, including Computer Helvetica): \$100. Corona Laser Printer and PCTeX: complete \$3395. System requirements: DOS 2.0 or later, 512K RAM, 10M hard disk. M/C, Visa accepted.

Personal
TEX
Inc

20 Sunnyside, Suite H
Mill Valley, CA 94941
(415) 388-8853 Telex 275611

Trademarks: PCTeX, Personal TeX, Inc.; TeX, American Mathematical Society; IBM PC and AT, IBM Corp; LaserWriter, Apple Computer, Inc.; Hercules Graphics Card, Hercules Computer Technology.

Circle no. 76 on reader service card.

**THE BEST Z80
ASSEMBLER ON
THE MARKET JUST
GOT BETTER!**

Z80ASM
NOW ONLY **\$49.95**

**DON'T ASK HOW OURS CAN BE SO FAST ...
ASK WHY THEIRS ARE SO SLOW!**

"... a breath of fresh air ..."

Computer Language, Feb. 85

"... in two words, I'd say speed & flexibility",

Edward Joyce, User's Guide #15

Now fully compatible with M80 in .Z80 mode with many extensions. Time & date in listing, 16 char. externals, plus many other features.

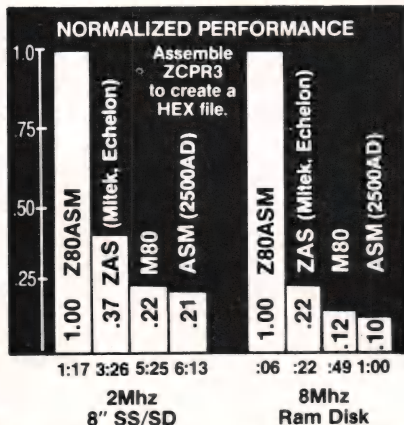
To order, or to find out more about our complete family of development tools, call or write:

SLR Systems

1622 N. Main St., Butler, PA 16001
(800) 833-3061, (412) 282-0864
Telex 559215 SLR SYS



C.O.D., Check or Money Order Accepted



SHIPPING: USA/CANADA + \$3 • OTHER AREAS + \$10
Z80 CP/M compatibility required.

Circle no. 78 on reader service card.

\$49.95 FMT \$49.95

Text Formatting System

FMT provides most of the features of the high-priced Text Formatters at our inexpensive price -- and it's easier to use, too! Note the features below:

- Easily configured to your printer. Configuration files for 20+ printer models are provided or generate your own.
- FMT gets the most from your printers by taking advantage of their special features, including condensed, double width, enhanced, double print, italics, elite, letter quality, multiple fonts, etc.
- Multiple modes and combinations of modes can be used on the same line or even in the same word.
- FMT works with your favorite editor!
- FMT uses meaningful mnemonic commands in the style of SCRIPT or ROFF (each command appears on its own input line), including commands for the various printing modes.
- No embedded control codes - you don't have to remember those strange escape/control sequences.
- FMT runs at the maximum speed your printer allows for each printing mode - graphics mode is not required.
- Standard formatting features provided, including headers and footers, automatic page numbering, text justification, tabs for table generation, and embedded files up to TEN deep.
- FMT automatically builds Table of Contents, List of Figures, and three level alphabetized Index.
- Detailed 100+ page manual profusely illustrated with examples.
- Works equally well with IBM-PC, TI-PC, IBM clones and look-alikes (PC-DOS/MS-DOS 128k). Also works with CP/M 8080 and Z80 systems with 64k.
- \$49.95 plus \$2.00 shipping and handling.

Specify system.

VISA and Master Card Accepted
Dealer Inquiries Welcome

TINY TEK, INC.

Route 1, Box 795
Quinlan, Texas 75474
(214) 447-3025

Circle no. 119 on reader service card.

Ram Disk Listing (Listing continued, text begins on page 84)

```

2564: 2564: 866 * INSTALL WAS SUCCESSFULL.
2564: 2564: 867 *
2564: A0 00 868 DISPVOL IDY #0
2564: B9 31 24 869 LDA VOLNAME,Y
2564: C9 00 870 CMP #EOL
2564: F0 09 2574 871 BEQ DV1
2564: 09 80 872 ORA #180
2564: 20 ED FD 873 JSR COUT
2572: C8 874 INY
2573: 4C 66 25 875 JMP DV0
2574: 2574: 876 *
2574: A0 00 877 DV1 LDY #0
2574: B9 5C 26 878 DV2 LDA VOLUME,Y
2574: C9 00 879 CMP #EOL
2574: F0 0C 258B 880 BEQ DV3
2574: C0 0E 881 CPY #14
2574: F0 08 258B 882 BEQ .DV3
2583: 2583: 883 *
2583: 09 80 884 ORA #180
2583: 20 ED FD 885 JSR COUT
2583: C8 886 INY
2583: D0 ED 2578 887 BNE DV2
2583: 2583: 888 *
2583: 20 8E FD 889 DV3 JSR CROUT
2583: 20 8E FD 890 JSR CROUT
2583: 20 8E FD 891 JSR CROUT
2583: 20 8E FD 892 JSR CROUT
2583: 20 3A FF 893 JSR BELL
2583: 60 894 RTS
2583: 2583: 895 *
2583: 2583: 896 *****
2583: 2583: 897 *
2583: 2583: 898 * PROGRAM TITLE
2583: 2583: 899 *
2583: 8D 8D 8D 900 MSG DFB CR,CR,CR
2583: 8D 8D 8D 901 DFB CR,CR,CR
2583: 20 20 20 20 902 ASC '
2583: 50 52 4F 44 903 ASC 'PRODOS RAMDISK DRIVER'
2583: 8D 8D 904 DFB CR,CR
2583: 20 20 20 20 905 ASC '
2583: 46 4F 52 20 906 ASC 'FOR FLASHCARD'
2583: 8D 8D 907 DFB CR,CR
2583: 20 20 20 20 908 ASC '
2583: 56 45 52 53 909 ASC 'VERSION 1.3'
2583: 8D 8D 910 DFB CR,CR
2583: 20 20 20 20 911 ASC '
2583: 41 4C 46 52 912 ASC 'ALFRED STEELE'
2583: 8D 8D 913 DFB CR,CR
2583: 20 20 20 20 914 ASC '
2583: 4A 55 4C 59 915 ASC 'JULY 1985'
2583: 8D 8D 8D 8D 916 DFB CR,CR,CR,CR
2583: 00 917 DFB EOL
2583: 2583: 918 *
2583: 2583: 919 *
2583: 2583: 920 *
2583: 20 20 20 52 921 VOLNAME ASC '
2583: 00 922 DFB EOL RAMDISK VOLUME NAME / '
2583: 2583: 923 *
2583: 2583: 924 *****
2583: 2583: 925 *
2583: 2583: 926 * CLEAR DIRECTORY FLAG
2583: 2583: 927 * SET FLAG FOR NO CLEAR
2583: 2583: 928 *
2583: 01 929 CLEAR DIR DB #01
2583: 2583: 930 *
2583: 2583: 931 *****
2583: 2583: 932 *
2583: 2583: 933 * MLI WRITE PARAMETER LIST
2583: 2583: 934 *
2583: 03 935 MLIWRITE DB #03
2583: 50 936 DB #50
2583: 57 26 937 DW BUFFER
2583: 02 00 938 WRITEBLOCK DW #0002
2583: 2583: 939 *
2583: 2583: 940 *****
2583: 2583: 941 *
2583: 2583: 942 * MLI READ PARAMETER LIST
2583: 2583: 943 *
2583: 03 944 MLIREAD DB #03
2583: 50 945 DB #50
2583: 00 20 946 READBUFF DW #2000
2583: 02 00 947 READBLOCK DW #0002
2583: 2583: 948 *
2583: 2583: 949 *****
2583: 2583: 950 *
2583: 2583: 951 * MLI READ BUFFER
2583: 2583: 952 *
2583: 2583: 953 * THE BYTES FROM BUFFER TO
2583: 2583: 954 * MIDBUFF IS THE RAMDISK VOLUME
2583: 2583: 955 * DIRECTORY HEADER. THESE BYTES
2583: 2583: 956 * HAVE TO BE WRITTEN TO THE
2583: 2583: 957 * RAMDISK AT BLOCK 2 BEFORE THE
2583: 2583: 958 * RAMDISK CAN BE USED BY PRODOS
2583: 2583: 959 *
2583: 2583: 960 * WE WILL ALSO USE PARTS OF
2583: 2583: 961 * THIS VOLUME DIRECTORY HEADER
2583: 2583: 962 * AS A CHECK TO SEE IF THE
2583: 2583: 963 * RAMDISK IS ALREADY FORMATTED
2583: 2583: 964 * IF SO THE USER CAN REFORMAT
2583: 2583: 965 * THE RAMDISK OR LEAVE IT ALONE
2583: 2583: 966 *
2583: 00 00 967 BUFFER DB #00,000
2583: 2583: 968 * LAST BLOCK
2583: 03 00 969 DB #03,000
2583: 2583: 970 * STORAGE TYPE/NAME LENGTH
2583: 2583: 971 DB #F3
2583: 5C: 2583: 972 * VOLUME NAME
2583: 52 41 4D 973 VOLUME ASC 'RAM'
2583: 00 00 00 00 974 DB #00,000,000,000
2583: 00 00 00 00 975 DB #00,000,000,000
2583: 00 0C 00 00 976 DB #00,000,000,000
2583: 2583: 977 * RESERVED
2583: 00 00 00 00 978 DB #00,000,000,000
2583: 00 00 00 00 979 DB #00,000,000,000
2583: 2583: 980 * CREATION DATE/TIME
2583: 00 00 00 00 981 DB #00,000,000,000
2583: 2583: 982 * VERSION/MIN VERSION
2583: 00 00 983 DB #00,000
2583: 2583: 984 * ACCESS BYTE
2583: 43 985 DB #43
2583: 2583: 986 * ENTRY LENGTH
2583: 27 987 DB #27
2583: 2583: 988 * ENTERIES PER BLOCK
2583: 0D 989 DB #0D
2583: 2583: 990 * FILE COUNT ON DISK
2583: 00 00 991 DW #0000
2583: 2583: 992 * BLOCK NUMBER OF BIT MAP
2583: 04 00 993 DW #0004
2583: 2583: 994 * TOTAL NUMBER BLOCKS ON DISK
2583: 00 02 995 DW #0200

```

(Continued on page 100)

Introducing Periscope II Professional Debugger and Break-out Switch



New Peri-
scope II
includes
a remote
break-out
switch that
does not
need its
own slot!

(Periscope is) "the best value in development tools currently on the market . . . the most essential element of my 'developer's toolbox'."

—Jeff Garbers

"Very powerful for debugging and testing . . . Better than Atron by far." —Wynn Bailey

The break-out switch "really sets Periscope apart from the typical software-only debuggers." Hung system or locked keyboard? Press the switch to get control!

Periscope's symbol support "beats the day-lights out of snooping through a map file and making notes". See high-level line numbers and source code, too!

"Feel right at home" in no time with commands that logically extend Debug's!

Periscope's speed makes other debuggers "look absolutely sluggish"! It's written entirely in assembler and uses DOS only for file access.

Has all the standard features plus:

- Debug with over 75 breakpoint options
- New!** Write your own breakpoint tests
- New!** Traceback
- New!** Do in-line symbolic assembly
- Debug using one or two monitors
- Recall command lines
- New!** Debug with high-level source code
- New!** Redefine windows while debugging
- New!** View text files while debugging
- Debug device drivers, non-DOS and memory-resident programs
- New!** Customize Periscope via user exits
- New!** Display 8087/80287 status
- New!** Use Periscope with an EGA

Periscope requires: IBM PC, XT, AT, or close compatible; DOS 2.0 & later; 128K RAM; 1 Disk Drive; 80-column Monitor.

Periscope II, break-out switch, manual, reference card and software . . . \$145!

Periscope I also includes the write-protected RAM board to protect crucial debugger code. It's just \$295!

The US Navy gets Periscopes from us . . . shouldn't you? Order today!

Order/Information Call Toll-Free:



800-722-7006



30-Day Money-Back Guarantee

Data Base Decisions • 404/256-3860
14 Bonnie Lane • Atlanta, GA 30328

Circle no. 130 on reader service card.

NEW!

ZBasic:

Finally, the simplicity of BASIC, yet faster than Turbo.

- Works the same on all micros
- Uses same commands—regardless of computer make
- Structured Basic—(spaghetti optional)
- Device-independent graphics (same graphic commands on all computers)
- 6 – 54 digits of precision (selectable by user)
- Built-in interactive editor and compiler—to compile and execute, just type "run."
- Choice of alphanumerics labels or line numbers
- Chaining with shared variables

**Not Copy Protected—
No Run Time Fees or
Royalties**
**One low price—No
Hidden Costs. Only
\$89.95 complete!**

••YES	ZBasic Interpiler 3.0	TURBO PASCAL 3.0	MBASIC compiler	True BASIC	BASICA Interpiler (IBM PC)
IBM and compatibles	•	•	•	•	•
Apple //e, //c (6502)	•	Not Avail.	Not Avail.	Not Avail.	Not Avail.
Macintosh	4th Qtr.	Not Avail.	Not Avail.	Not Avail.	Not Avail.
CP/M-80 2.2, 3.0	•	•	•	Not Avail.	Not Avail.
TRS-80 Mod I, III, 4, 4p	•	Not Avail.	•	Not Avail.	Not Avail.
Direct commands	•	Not Avail.	Not Avail.	•	•
Maximum scientific digits of accuracy (COS, SIN, ATN, LOG, EXP etc.)	6 to 54 selectable by the user	11 Binary BCD Not Avail.	16	16	6
Device Ind. Graphics (same commands all graphic modes and computers)	•	Not Avail.	Not Avail.	Not Avail.	Not Avail.
SAME File commands all computers?	•	Not Avail.	Not Avail.	Not Avail.	Not Avail.
STRUCTURED: Labels, Functions, LONG IF etc.	•	•	Not Avail.	•	Not Avail.
Same editor commands all versions/computers	•	•	Not Avail.	Not Avail.	Not Avail.
Sieve benchmark (Byte January 1983, 10 iter's)	13.7 sec.	14.1 sec.	14.9 sec.	261 sec.	2190 sec.
Shell-Metzner SORT (Sybex-BASIC for Scientist's and Eng. 2,000 5 char. strings)	19 sec.	28 sec.	71 sec.	194 sec.	2700 sec.
Executable Machine Lang. & approx. File size	12k	12k	32k	Not Avail.	Not Avail.
PRICE with BCD BCD=No rounding errors)	89.95	109.95	450.00	Not Avail.	Not Avail.
PRICE without BCD	89.95 BCD FREE	69.95	395.95	149.95	Comes with computer

ZBasic is a trademark of ZEDCOR Inc. © 1985 IBM is a registered trademark of International Business Machines Corp. Apple //e, //c are trademarks of Apple Corp. Inc. Macintosh is a trademark licensed to Apple Corp. Inc. PM-80 is a registered trademark of Digital Research Inc. TRS-80 is a registered trademark of the Tandy Corp. TURBO-PASCAL is a registered trademark of Borland Int. MBASIC is a registered trademark of Microsoft Corp. True BASIC is a registered trademark of Addison-Wesley Publishing Co. BASICA is a registered trademark of International Business Machines Corp. All benchmarks and accuracies apply to standard IBM PC with 8088. Other computers and/or CPU benchmark speeds, accuracies and object code file sizes will vary depending on the computer clock speed, operating systems and other factors beyond our control.

ZBasic™

The lightening-fast BASIC
from Zedcor

To order use this coupon or call (orders only): 1-800-482-4567
INQUIRIES: (602) 795-3996

(Name) _____
(Address) _____ (Apt.) _____
(City) _____ (State) _____ (Zip) _____
(Day Phone) _____

CREDIT CARD—MASTERCARD/VISA/AMEX/C.O.D.

CREDIT CARD EXPIRATION DATE:

Arizona Residents add 5% Sales Tax. C.O.D. add additional \$5.00 per order. Orders outside U.S. and Canada add \$20.00 per shipment for postage (U.S. Currency only)

Mail to:
ZEDCOR INC
4500 E. Speedway, Ste. 93
Tucson, AZ 85712

\$89.95 for each version

- ☐ IBM PC & Compatibles
- ☐ Apple //e, //c
- ☐ CPM-80
- ☐ KAYPRO Graphics
- ☐ TRS-80 (Model 1.3)
- ☐ TRS-80 (Model 4)
- ☐ Macintosh (4th QTR '85)
- ☐ DEVELOPERS—BUY ALL 6—
\$399.75 (\$449.75 VALUE)

SAVE \$140.00

U.S. shipping \$5.00

per order (\$10.00 C.O.D.) \$ _____

TOTAL

\$ _____

Circle no. 160 on reader service card.

Ram Disk Listing (Listing continued, text begins on page 84)

```


2682: 996 * REST OF BUFF MORE THEN NEEDED
2682: 0200 997 MIBUFF DS 512

234B ABORTMSG 23F9 BE0 2409 BE1 FF3A BELL
2235 BITMAP D2 BLOCK1 D3 BLOCK2 ? D4 BLOCKNUM
D5 BLOCK4 D6 BLOCK5 ? D7 BLOCK6 46 BLOCKNUM
2235 BM2 44 BUFFADR 23F1 BUFFERERROR 2657 BUFFER
2118 C1 211C C2 2124 C3 2126 C4
2201 CB1 2215 CB2 2218 CB3 2277 CD0
2286 CD1 2295 CD2 22AA CD3 22BA CD4
21E5 CE 2250 CHECKDEV 222B CL1 2228 CLEARBUFF
264A CLEARDIR 21E2 CLEAREND 42 COMMAND ?210C COMPUTE
2169 CONT FDED COUT FDSE CROUT 8D CR
22D9 DA1 22E9 DA2 22D1 DEVBABORT BF1A DEVADR51
BF31 DEVCNT BF32 DEVLST 06 DIRBLOCKS 22EF DIRFOUND
2564 DISPVOL 2566 DV0 2576 DV1 2578 DV2
2588 DV3 218F ENDSSD 00 EOL 217E EX1
218E EX2 217A EXIT BEF5 GETBUFF 2541 GETKEY
2544 GX1 FC58 HOME C000 KEY 2190 MAIN
2682 MIBUFF 2648 ML1WRITE BF00 ML1 21651 ML1READ
21B3 MN1 259B MSG 24B3 NOBUFF DOA2 NODEV
22CB NODIR 2186 R1 48 RAMBANK 2655 READBLOCK
?2653 READBUFF ?2156 READBYTE 01 READ 21B4 RELOCATE
2104 S1 08 SAVEBYTES 21C8 SD1 213C SETBLOCK
21C1 SETDEV CDD0 SETVECTOR 02 SSDCODE COD1 SSDHI
214C SSDIO CDD0 SSDLOW 2100 SSDSTART C010 STROBE
4A TEMP ? 00 TEMPZERO 2554 T11 2543 T12
254F TITLE ? 43 UNITNUMB 2631 VOLNAME 265C VOLUME
BE00 WARMDOOS 240F WHEREMSG 2397 WHERETO ? 02 WRITE
264F WRITEBLOCK 2161 WRITEBYTE 21EF WRITEDIR 239C WT0
23AC WT1 23CB WT2 23DA WT3 23E8 WT4

** SUCCESSFUL ASSEMBLY = NO ERRORS
** ASSEMBLER CREATED ON 14-JUN-83 09:55
** TOTAL LINES ASSEMBLED 997
** FREE SPACE PAGE COUNT 87

```

End Listing



Experteach™

The Complete and Comprehensive Intellware™ Laboratory for Expert System Concepts on the IBM Personal Computer.

Experteach Includes:

- Comprehensive Introduction to Expert System Concepts.
- On-line Tutorial Describing the Operation of Expert Systems.
- Lisp Based Expert System Tools with Source Code.
- Prolog Based Expert System Tools with Source Code.
- dBASE II™ Based Expert System Tools with Source Code.
- Pascal Based Expert System Tools with Source Code.
- Complete Lisp Interpreter for the IBM PC®.
- Complete Prolog Interpreter for the IBM PC.
- Comprehensive Case Studies of Several Major Expert Systems.
- Comprehensive bibliography on Expert Systems.

Experteach is a comprehensive guide to Expert System technology consisting of a uniquely integrated collection of Expert System tutorials, case studies, on-line teaching programs, Expert System building tools with source code and Artificial Intelligence languages. Experteach is based on extensive experience in teaching Expert System concepts in association with IEEE, ACM and the Continuing Education Institute.

Experteach introduces you to Expert System technology by allowing you to build Expert Systems and to experiment with a variety of Artificial Intelligence tools and languages on the IBM PC.

Experteach includes eight rule-based Expert System shells with source code implemented in Lisp, Prolog, dBASE II and Pascal. Each language has been used to implement both a forward chaining and a backward chaining Expert System shell with a built-in rule editor, inexact inference and how & why explanation facilities.

Experteach includes a comprehensive Lisp interpreter and a complete Prolog interpreter with DEC-10 Prolog syntax. Experteach requires only 256K of memory.

Intellware, Inc., 4676 Admiralty Way Suite 401 Marina del Rey, CA 90291 (213) 827-1334

- Introduction to Expert System Concepts, On-line Tutorial, Case Studies, Bibliography and Pascal Based Tools \$99.00.
- Lisp, Prolog or dBASE II Based Tools \$99.00 each.
- Complete Experteach System with Lisp and Prolog Interpreters. \$475.00 Check, Money order, Visa or Mastercard. \$9.00 for postage and handling. California, 6.5% tax.

Intellware, Inc.
Experteach and Intellware are trademarks of Intellware, Inc. IBM PC is a registered trademark of IBM. dBASE II is a TRADEMARK of Ashton-Tate.

Circle no. 65 on reader service card.

CP/M™

UPGRADES - FROM \$49.95!


ConIX Operating System: Adds more UNIX™-like features to any 48K+ CP/M-80 or compatible micro then ever before. It's a total upgrade for CP/M! Includes I/O Redirection and Pipes (uses memory or disk), perfected User Areas, Command and Overlay Path Searching, Auto Screen Paging, 8Mb Print Buffering, 22 new SysCalls, Function Keys, "Virtual" disk system, Archiver (saves over 50% disk), extensive command language, 300+ variables, 100+ commands, and even more! Uses 1/2K TPA, 0-27K disk minimum. 60-second install time!

ConIX Library I - XCC Utilities: Over 20 utilities written in the ConIX shell language, including hierarchical directories, interactive debugger, move/copy/link multiple files, print files with pagination, review disk files for deletion, unerase disk with stats, full-screen TYPE, and more. Source code included!

"(ConIX) goes a great deal farther in flexibility than any other CP/M shells. . . this system has all of its competitors beat." *As reviewed in Computer Language, June 1985.*

- ConIX Basic System: List \$99.95, *Special - only \$49.95*
- ConIX Programming System: List \$99.95, *Special - only \$49.95*
- Both Systems 100% Complete: List \$165, *Special - only \$79.95*
- ConIX Pull-Down Menu: List \$35, *FREE with any ConIX system*
- ConIX Library I: List \$49.95, *with any ConIX system only \$34.95*

Prices include manual, 8" disk, and user support. 5 1/4" conversions 48TPI \$6, other \$10. Shipping: \$4.50 UPS, \$10 Canada, \$15 overseas. NY residents add sales tax. Dealers wanted - special incentives!



Computer Helper Industries Inc.
P.O. Box 680 Parkchester Station, NY 10462
800-628-2828 X513 24hr. order hotline - COD only
212-652-1786 M-F:10-6 sales/technical staff

UNIX: AT&T Bell Labs, CP/M: Digital Research, ConIX: Computer Helper Ind.

Circle no. 129 on reader service card.

No source code for
your REL files?

REL/MAC

converts a REL file in the Microsoft™ M80 format to an 8080 or ZILOG™ Z80 source code MAC file with insertion of all public and external symbols.

- REL/MAC makes MAC source files
- REL/MOD lists library modules
- REL/VUE displays the bit stream
- REL/PAK includes all of the above
- 8080 REL/MAC demo disk \$10.00

REL/PAK for 8080 only \$99.95
REL/PAK for Z80 & 8080 \$134.95
on 8"SSDD disk for CP/M™ 2.2

Send check, VISA, MC or C.O.D. to



MICROSMITH
COMPUTER TECHNOLOGY

P.O. BOX 1471 ELKHART, IN 46515

1-800-622-4070

(Illinois only 1-800-942-7317)

Circle no. 103 on reader service card.

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)

OUTSIDE OKLAHOMA: NO SALES TAX.

V20 CPU JPD70108D-8 \$16.00
8087-2 Math Coprocessors 140.00

DYNAMIC RAM

256K	64Kx4	150 ns	\$8.45
256K	256Kx1	120 ns	3.25
256K	256Kx1	150 ns	2.47
128K	128Kx1	120 ns	5.20
64K	16Kx4	150 ns	2.75
64K	64Kx1	150 ns	1.00

EPROM

27C256	32Kx8	250 ns	\$9.99
27256	32Kx8	250 ns	4.75
27C64	8Kx8	150 ns	3.99
2764	8Kx8	250 ns	2.25

STATIC RAM

6264LP-15	8Kx8	150 ns	\$3.25
-----------	------	--------	--------

OPEN 6½ DAYS: WE CAN SHIP VIA FED-EX ON SAT.

MasterCard/VISA or UPS CASH COD

Factory New, Prime Parts JPD

MICROPROCESSORS UNLIMITED, INC.

24,000 S. Peoria Ave., (918) 267-4961

BEGGS, OK. 74421

Prices shown above are for Nov. 18, 1985

Please call for current prices. Prices subject to change. Please expect higher or lower prices on some parts due to supply & demand and our changing costs. Shipping & insurance extra. Cash discount prices shown. Orders received by 6 PM CST can usually be delivered to you by the next morning, via Federal Express Standard Air @ \$6.00, or Priority One @ \$13.00!

ORDERS BY
TH. 5:00 AM
FR. 9:00 AM
NO EXTRA
COST ON
FED-EX SAT
DELIVERY

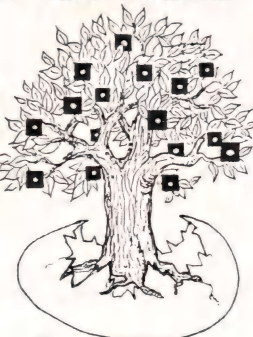
640 KYTE MOTHERBOARD KITS: IBM PC/XT
Zenith 150 & 100, Compaq Portable & Plus: \$65.00

QUANTITY ONE PRICES SHOWN
NEW! 100 ns 256Kx1 D-RAM @ 5.15

Circle no. 105 on reader service card.

Tree Shell

A Graphic
Visual Shell for
Unix/Xenix
End-Users and
Experts Alike!



Dealer
inquiries
welcomed.

COGITATE

"A Higher Form of Software"

24000 Telegraph Road
Southfield, MI 48034
(313) 352-2345

TELEX: 386581 COGITATE USA

Circle no. 81 on reader service card.

ASSEMBLERS

We will support you on over 20 UNIX
and Xenix based machines. Targets:

Fairchild	F8/3870
Hitachi	6301, 6305, HD64180
Intel	8041, 8048, 8051, 8080 8086 family
Motorola	6800, 6801, 6805, 6809 68HC11, 68000 family
NSC	800
RCA	1802
Rockwell	6502/65c02
Texas Inst.	TMS7000
Zilog	Z8, Z80

UNIWARE™

SOFTWARE DEVELOPMENT SYSTEMS, INC.

3110 Woodcreek Drive

Downers Grove, IL 60515 U.S.A. (312) 971-8170

England: Unit C, Ltd. (0903) 205233

UNIX is a Trademark of AT&T Bell Labs
Xenix is a Trademark of Microsoft

Circle no. 115 on reader service card.

HISOFT

HIGH QUALITY CP/M
SOFTWARE

HiSoft has been selling Z80 CP/M software in Britain and Europe for over 4 years. Now we'd like to introduce you to our range of programming languages:

HiSoft Devpac: Z80 assembler/editor/
debugger

HiSoft C: Kernighan/Ritchie
implementation

HiSoft Pascal: fast, standard compiler
All at \$69 inclusive each.

These programs are also available for other Z80 machines including Timex 2068.

Call or write for full technical details and press commentaries, or order from:



HISOFT

180 High St. North
Dunstable LU6 1AT
ENGLAND
01144 (582) 696421

Circle no. 134 on reader service card.

UNIVERSAL CROSS-REFERENCER

• IT WORKS WITH **ALL**
LANGUAGES

BASIC, C, PASCAL, FORTRAN, COBOL,
ASSEMBLER, dBASE... you name it.

• IT HANDLES standard languages,
extended languages & exotic languages.

• IT CONFIGURES to your com-
piler, interpreter, or assembler — ALL
BRANDS — ALL VERSIONS.

INTRODUCTORY
PRICE **\$39.95**

\$3.00 S/H - MC/Visa/Check - 6% Texas Sales Tax
For the IBM PC, XT & compatibles. DOS 2.0+

DALSOFT SYSTEMS

3565 High Vista - Dallas, TX 75234
(214) 247-7695

Circle no. 86 on reader service card.

THE HAMMER Software Tools in C

"I have already saved weeks of coding... thank
you for providing such a useful tool..." G.T.

Let The HAMMER Library of over 150 routines save you
valuable development time and effort:

LIBRARY FUNCTIONS

- Multi-level 123-like MENUS

• DATA ENTRY

- MULTI-FIELD mode for Full-Screen data entry
- Single-Field mode for individual fields
- Data Verification
- Full Editing within each field
- Strings, dates and fixed decimal numbers
- "Option" fields force user to pick from a given set

• SCREEN MANAGEMENT

- cursor positioning
- display boxes & tables
- full attribute control
- scrolling and clearing
- Date/time/string conversions
- BIOS access/pattern matching/and more

SUPPORTS: Microsoft C 3.00 • C/C86 • M W C86 •
DeSmet C • Lattice

INCLUDES source code and manual \$195 plus shipping
VISA/MC accepted

O.E.S. SYSTEMS

1906 Brushcliff Rd. • Pittsburgh, PA 15221
412/243-7365

Circle no. 137 on reader service card.

FORTH

and Assembler for

65802/816

16 bit Processing for Upgraded 6502 Systems

Forth, Assembler, and Debugger for

65 Series Processors

On a VAX or PC*

Starlight Forth Systems

15247 N. 35th St
Phoenix, AZ 85032
(602) 992-5181

*65802 on Standard 6502 systems

Circle no. 142 on reader service card.

C Users'
Group

Over 60 volumes of public
domain software including:

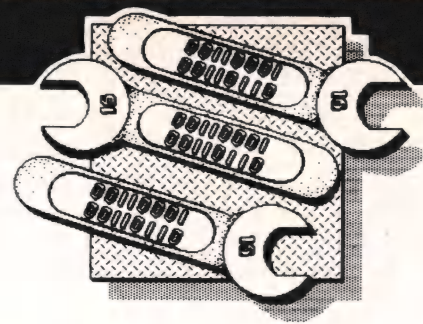
- compilers
- editors
- text formatters
- communications
packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

Post Office Box 97
McPherson, KS 67460
(316) 241-1065

Circle no. 63 on reader service card.



by Ray Duncan

The programs published in this month's column are available for downloading from the Laboratory Microsystems RBBS at (213) 306-3530 (300 baud or 1200 bps).

Microsoft Assembler

I never cease to be astonished at the flurry of letters that results every time we publish something about the Microsoft Macro Assembler.

J. C. Hoisington of Morgan Computing (publisher of Trace86 and Professional Basic) writes: "We switched from IBM to Microsoft at Version 2.0. That means we didn't get SALUT but we did get to upgrade to Version 3.0 early this year for \$75.00. Version 3.0 finally corrects many of the errors (and ridiculous restrictions) of earlier versions. The most useful to us is that they finally allow

```
lods byte ptr es:[si+0]
```

"which assembles as 26AC. I still found many bugs in Version 2.0, but 3.0 seems to have very few glitches."

David L. Rabbers, of Seattle, Washington, writes: "Regarding the errors [published in previous columns] two of them are yours, not the assembler's. Specifically, any constant ending in b or B is always treated as binary, and any constant ending in d or D is always treated as decimal. This is true regardless of the current radix setting. So, if you set the radix to 16, you must see to it that hex constants ending in b, B, d, or D are followed by an h or H. A little inconvenient, but how else could you specify binary or decimal constants in the midst of radix 16?"

"One feature of Microsoft MASM Version 3.0 is a new linker that supposedly supports simple overlays.

Unfortunately, there is no hint that I can find on how to use this feature. If you know how, or discover how, *please* publish it." Good point. My manuals don't breathe a word about the construction of overlays. Anyone out there have some inside information on this subject?

Robert A. Blair, of Walnut, California, made the same point about radix overrides and went on to say: "This assembler should be used in Computer Science courses in college as an example of the wrong way to design a compiler; it is the worst I have ever seen. If you thought that Version 1 had a zillion bugs, wait until you use Version 2 for a while.

"I have to define my data before I use it (I thought I only had to do that in Pascal) or explicitly tell the compiler where it is to avoid phase errors. A multi-pass compiler should be able to figure out where the data is located without my having to tell it. The accompanying assembly listing shows a few errors that I know about. The 80286 PUSH immediate instruction generates the wrong length for some values. It will generate an instruction that requires two bytes of immediate data and only inserts one byte, or generates an instruction that requires one byte of data and inserts two bytes. The OFFSET operator causes link errors occasionally. The MASK used with HIGH does not work ... These are just the tip of the iceberg." Robert's contribution accompanies the column as the listing (page 106).

Converting to DOS 3

Dan Daetwyler is a regular correspondent to *Doctor Dobb's* and always has something interesting to say. This month he shared some of his experiences with the latest version of DOS: "When DOS 2.0 introduced the

concept of a file handle, I looked it over fairly carefully, and then decided we were being shown 'the way to go.' I went. It was a nice concept, and effectively the number of file handles that could be used at one time was unlimited. They made it so you could not use more than 99 handles at once—a good, typical effort by programming to cope with the concept of infinity on a finite machine. This was all well and good, since no one was going to have more than 99 files going on a PC.

"Then along came the new Version 3.x DOS. Gee, it just *had* to be better (and bigger), and all that good stuff. I read the new features section, and there *were* a couple of goodies. I devoured them carefully, and put them to immediate use. I bought Version 3.0, even though each time I swear I'll wait for the x.1 release—I get so tired of fighting MS bugs. Still, I bought it, and the Technical Reference, and really thought I read the documentation thoroughly. Along came Version 3.1, for which I breathed a large sigh of relief, since I could now stop fighting DOS bugs and worry about my own. And then I hit the thing with one of my moderately large complex systems. And bounced like a rubber ball.

"Buried in the details of the FILE statement in the CONFIG dissertation is a neat little statement that they've arbitrarily reduced the number of active handles to 20! The constraint is for a process, which is interestingly undefined, but you can assume it's a DOS job or task. The horrible part is, they meant it!

"So 20 isn't too bad a constraint. There won't be too many people who want to exceed that limit. Well, on the same page there's a note that five of the handles are grabbed by DOS.

That's very true. So now you're down to 15. Still a reasonable number? Not in my estimation. Let's follow through on a reasonably complex data base problem. We've got several—let's say five—master files, each of which is indexed. The three primary files have double indices, while the less important two have only a single index. Five plus three times two plus two gives us thirteen. Still safe? Not really. As you're aware, the only way you can force a buffer purge in DOS is to pseudo-close a file. Doing this with a duplicate handle permits you to avoid the killing performance of a re-open. Let's be stingy. We'll only use one handle for this purpose. We've still got one left. However, I've yet to write a system that doesn't have at least three and sometimes half a dozen supporting files for archives, output, etc. The net result is that you end up short about five handles, even on a relatively small system."

Dan's idea of a "relatively small system" would give many of us nightmares. Note how Dan offhandedly supplies the reason for the existence of that mysterious DOS function 45H called DUP, whose "explanation" in the DOS Technical Manual is "Purpose: returns a new file handle for an open file that refers to the same file at the same position." I had always been perplexed by this function, and the Unix manuals weren't any more helpful than the DOS manual. By the way, the 5 handles preassigned by DOS to the Standard Devices aren't really lost. You can always close them, and then reuse them for some other file or device.

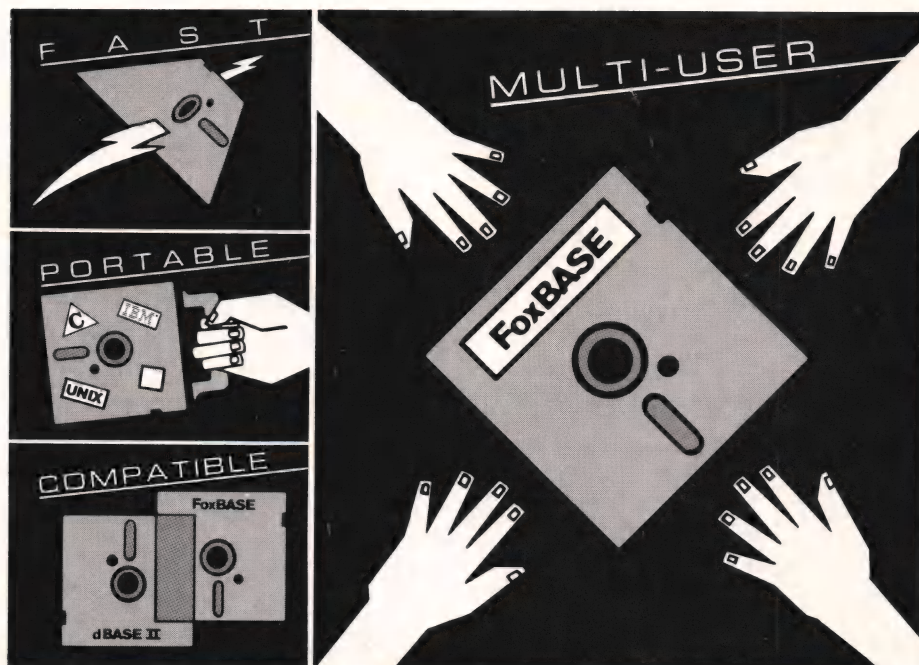
Dan writes on: "My current effort involves a minimum of 33 handles, and even that makes me uncomfortable. Goodbye to DOS Version 3.x. Except I can't! I can't really lock a customer into the old DOS. Especially with multi-tasking 'right around the corner.'

"If you think this tale has a happy ending, you're wrong. Let me tell you some of the things I've tried. You can set up a resident task, split the data base, and let the resident task own part of it. The catch is that if you try to communicate with the resident

task, you're running under the main task control structure, and DOS will tell you that the access isn't authorized. I haven't figured out how DOS knows, as yet, so I can't figure out how to switch task structures. The traditional use of a user interrupt vector doesn't work. DOS senses this and tells you again that the access is not authorized. Setting the file up with the proper attributes for a shared resource, and loading SHARE, doesn't help. I guess I could fight through a disassembly of DOS and find out what they're doing, but I

don't want to spend the rest of the summer on this.

"I started to drop back to my own FCBs, and drop the handle concept. There's no limit to the number of files I can open, then, but then I'm back to the problems of the pre-handle DOS, and I can't see any way to purge the buffers. I can't really see exposing my customer to loss of data base when the power drops unexpectedly. I can't actually give him total integrity, but using the pseudo-close purge gets the risk factor down to the noise level. I've three of the complex database



FoxBASE. The DBMS That Bridges The Gap Between Single-user And Multi-user.

Unsurpassed Program Development Speed.

FoxBASE™ uses a state-of-the-art B+ Tree index structure for quicker, more efficient data access. A sophisticated virtual storage technique becomes an invaluable timesaving device as it works to insure that frequently referenced programs are retained in memory in compiled form. What's more, FoxBASE provides automatic 8087/80287 math coprocessor support for ultrarapid program execution speed—as much as six times the speed of dBASE II®.

Highly Portable.

FoxBASE is much more than a relational database management system. Written in C, FoxBASE is an extremely portable interpreter/compiler. Now you can port from one machine or operating system to another without changing your applications. And this portability protects your investment in programs by insuring their use in future machine and operating system environments.

dBASE II Compatible.

FoxBASE is both source language—including full macro usage—and data file compatible with the dBASE II database language. This means your existing dBASE II databases can be used unchanged. Furthermore, it puts thousands of public-domain and commercially available dBASE II programs at your disposal.

Available Under Multi-user Systems.

Our multi-user versions of FoxBASE feature many additional enhancements. Like automatic file-locking and record-locking capabilities. Use of termcap, so FoxBASE can run on virtually any terminal. And, with some versions, a two billion record file capacity.

Multi-user Versions:


Xenix® \$995. MultiLink™ \$995.

IBM-PC NET™ \$995.

Single user Versions:

MS/PC-DOS™ \$395. AOS/VS \$995.

Don't be outfoxed by the others.
Call or write Fox Software today.

FoxBASE™ 

FOX SOFTWARE, INC.

27475 Holiday Lane, Perrysburg, OH 43551
419-874-0162

FoxBASE is a trademark of Fox Software, Inc. dBASE II is a registered trademark of Ashton-Tate. Name is a registered trademark of Microsoft Corp. MultiLink is a trademark of The Software Link, Inc. IBM PC NET is a trademark of IBM Corp. MS-PC-DOS is a trademark of Microsoft Corp.

Circle no. 94 on reader service card.

9 TRACK TAPE CONTROLLERS AND 1/2" TAPE SUBSYSTEMS

MODEL TC-PC

TC-PC is a high performance 9-track tape controller for the IBM-PC with these important features:

- Reads and writes industry standard 1/2-inch tape
- Compatible with most formatted tape drives
- Standard 8-bit parallel recording with parity, and read-after-write verification
- Switch selectable I/O address (four contiguous ports required for operation)
- Maximum data transfer rate of 192,000 bytes per second
- Record length from 1 to 65,535 bytes
- Supports up to 8 tape transports
- Jumper selectable DMA channel
- Modes: PE and NRZI at 800, 1600, 3200 and 6250 bytes/inch
- Installable device drivers allow creation of application programs which run under IBM XENIX and MS-DOS
- Operates with IBM-PC and -XT; Compaq Portable; Zenith PC-150; Sperry PC; the Leading Edge Computer, and other 100% IBM-PC compatible equipment.

MODEL TC-50

TC-50 offers all the standard features of the TC-PC with these additional enhancements:

- Maximum data transfer rate of 400,000 bytes/second; 904,000 bytes/second with memory option
- Operation with a wider range of IBM-compatible machines, including IBM-AT; Compaq Desk Pro; ATT 6300 and others

A variety of software utilities is supplied as part of the TC-PC and TC-50 packages, including:

• DEPOT (Data Exchange Program with Optional Translation)

DEPOT provides a means to transfer data between system disk and magnetic tape, allowing:

- Data interchange from tape to disk, and disk to tape
- Conversion from ASCII to EBCDIC, and vice versa
- Positioning to arbitrary location prior to data read
- Specification of record length and block factor when writing from disk to tape; allows deblocking when reading from tape to disk
- Multiple operations to be specified from a command file

• TAU (Tape Archive Utility)

- Provides individual file backup and restore
- Allows use of MS-DOS wild cards such as *.*.*
- Provides disk drive selections for I/O
- Changes pathname selections from within TAU
- Provides data encryption for security

WARRANTY

All Overland Data products carry a 30-day unconditional money-back guarantee, and are warranted for one year, parts and labor.

OVERLAND DATA, INC.

5644 Kearny Mesa Road #A
San Diego, CA 92111
Tel. (619) 571-5555

XENIX and MS-DOS are Registered Trademarks of Microsoft Corp.
IBM-PC/AT/XT are Registered Trademarks of International Business Machines Corp.
Compaq Portable and Compaq Deskpro are Registered Trademarks of Compaq Corp.
ATT 6300 - AT&T Information Systems Corp.
Sperry PC - Sperry Univac Corp.
Zenith PC-150 - Zenith Data Systems
Leading Edge is a Registered Trademark of Leading Edge Products, Inc.

Circle no. 73 on reader service card.

Vitamin C

SCREEN I/O FEATURES INCLUDE:

- dBase-like atsay(), atget(), readgets()
- Input formatting via picture clause
- Unlimited input validation
- Individual field color/attribute control
- Field specific help messages
- Status line for added field specific prompts
- Right-to-left numeric input with floating dollar signs, asteriks, and commas
- Insert/delete, next/previous field, etc
- Current field highlighting
- Fast, easy, and bulletproof

More than just another function library!

A well planned, fully integrated programming system complete with pull down menus & user help system!

Vitamin C • \$149.95

Includes more features than there is room to mention, all source code, manual, tutorial and sample programs. No royalties on your applications.

For most MS-DOS C compilers! Soon for UNIX / XENIX! Include \$3 shipping, \$6 for second day. Phone orders with MC/Visa welcome!

Prices & features subject to change without notice. Call for other products.

A Healthy Supplement For C Programmers

PERFECT WINDOWS

- Automatic interface with all screen I/O functions
- Full overlay and restore
- Multiple virtual screens
- Full collision protection
- Automatically keeps I/O within window boundaries
- Grow, shrink and move
- Word wrap & margins
- Print to or scroll any window without collision even if it is overlayed by another!
- Much, much more!

CREATIVE PROGRAMMING

(214)243-6197

Box 112097

Carrollton, Texas 75011-2097

Circle no. 82 on reader service card.

systems out now, and in use on about fifty machines. We're semi-rural, and the power fluctuation during thunderstorms has to be seen to be believed, yet I've never (knock wood) lost a database from that source, since I started using the pseudo-close.

"So right now, I'm stuck. I can't move two of my systems to DOS 3.x, and the new one isn't even in the ball park . . . So this tale ends with questions. Do you know anything of help in this area?"

Actually, even the DOS 2.1 manual (pages 4-10) said that the maximum number of handles that a process could have concurrently open was 20, but that the maximum for the system was 99. This apparently includes background tasks such as the print spooler. Perhaps they just didn't enforce the process maximum. The manual for DOS 3.1 specifies the maximum for the system can be in the range 8-255, but again the maximum per process is 20. For those of you who are still using FCBs for whatever reason, be warned that DOS 3.1 puts some new and fairly strict restrictions on the use of FCBs when the file sharing module is also loaded. The default maximum number of FCBs that can be simultaneously open is 4. If you try to open more than your maximum, DOS unilaterally closes the least recently used FCB. You can increase the maximum allowed number of simultaneously open FCBs by a command FCBS=nn in the CONFIG.SYS file, just as FILES=nn works for handles. This is an omen, I believe, that FCBs are as doomed as the dinosaurs.

Microsoft and IBM have issued particular warnings about use of FCBs in the networking environment. If you use an FCB to access a file and fail to close it (a very common thing to do when you are using the file for read only), the system can bog down due to excessive connections accumulating across the network. Also, they remark that moving FCBs around in memory after they are opened can cause problems.

Dan proceeds: "Tale two is slightly more amusing. When I got my AT, and then started hearing all the terror stories about the hard disk, I consid-

ered myself very fortunate when I didn't encounter any . . . for a while. Then I had to load a 5 Mbyte database, and my trouble started. After several days of frustration, I nailed my friendly dealer to the wall, and IBM emergency shipped a replacement hard disk, which made life a lot better. In the interim, I did several interesting things. I put a trap in the program so that any time the I/O failure occurred, I'd get control. I was running under DOS 3.0, of course. The first thing I discovered was that when the disk glitched, the error code that DOS gave me back had no relation to disk I/O at all. In fact, DOS kept insisting that I was 'out of memory.' On a half megabyte machine with one small application?—not bloody likely.

"The next thing I discovered was that if I simply tried the operation again, by looping back to the interrupt, 99% of the time the thing worked happily on the second try. Very interesting! Then I discovered by accident that I was running with `VERIFY=OFF`. So I changed to `VERIFY=ON`, and got a whole new set of behavior patterns. Different error codes, and the fail ratio went way down. Running in this mode, the problem seemed to be occurring when DOS encountered something it thought was a bad track, so I off-loaded my 20 Mbyte disk, and reformatted. Oh boy. The plot thickens like glue. I could run `FORMAT` repetitively, and every time I ran it I would find a different set of 'bad tracks.' There didn't even seem to be a consistent pattern. I believe it did have a relationship to something, but as near as I could detect, it was the phases of the moon, or the position of Jupiter.

"I won't bore you with more of the experiments that cost me about ten days of time, but I will tell you the punch line. We all heard about the hard disk problems in all of the trade journals. IBM replaced hard disks left and right, and the disk vendor was the bad guy. Very interesting, since, when I got Version 3.1, all the problems disappeared. Now DOS gives back the correct error codes. Now `FORMAT` is at least consistent."

Dan's second tale is the latest of a

raft of mutually exclusive explanations of the mysterious AT hard disk problems. We have heard that the original hard disk was no good because the vendor tried to ramp up production too fast (and a recent review of the CMI hard disk showed it to be much more shock-sensitive and error-prone than its competitors); that the `FORMAT` program had a bug that prevented it from actually checking the second half of the 20 Mbyte hard disk; that the disk controller had design or component

problems; and so on. Throughout most of this time, the official IBM press flacks insisted that everything was wonderful and there was no AT disk problem at all. It certainly makes one apprehensive about the position we'll all be in if IBM manages to knock off all its competitors in the PC market.

DDJ

(Listing begins on next page)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 197.

It's 3 AM!



Do you know where your bugs are?

This C programmer is finding his bugs the hard way...one at a time.
That's why it's taking so long. But there's an easier way. Use

PC-Lint

PC-Lint* analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. PC-Lint resembles the Lint that runs on the UNIX* O.S., but with more features and some awareness of the 8086 environment.

- Full K&R C
- Supports Multiple Modules—finds inconsistencies between declarations and use of functions and data across a set of modules comprising a program.
- Compares function arguments with the associated parameters and complains if there is a mismatch or too many or too few arguments.
- User-modifiable library description files for most major compilers.
- All warning and information messages may be turned on and off globally or locally (via command line and comments) so that messages can be tailored to your programming style.
- All command line information can be furnished indirectly via file(s) to automate testing.
- Use it to check existing programs, programs about to be exported or imported, as a preliminary to compilation, or prior to scaling up to a larger memory model.
- All one pass with an integrated pre-processor so it's very fast.
- Has numerous flags to support a wide variety of C's, memory models, and programming styles.
- **Price: \$139.00 MC, VISA**
(Includes shipping and handling) PA residents add 6% sales tax. Outside USA add \$10.00
- Runs under MS-DOS* 2.0 and up, with a minimum of 128Kb of memory. It will use all the memory available.

Trademarks: PC-Lint (Gimpel Software), UNIX AT&T, MS-DOS (Microsoft).

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426

(215) 584-4261

16-Bit Listing (Text begins on page 102)

```

1                                     page    55,132
2                                     title    Microsoft Assembler Bugs
3                                     .286C
4
5                                     ; Some IBM PC Macro Assembler 2.0 "features"
6                                     ; contributed by Robert Blair
7
8      0000                          foo      segment para public 'CODE'
9                                     assume   cs:foo
10     0000                          foos     proc    far
11     0000  E9 010B R                jmp      foobar
12                                     ; this bug causes link errors if
13                                     ; the offset is larger than 256.
14                                     ; the error is
15                                     ; 'fixup offset exceeds field width'
16     0003      02 [                dw      2 dup (offset foobar)
17                                     010B R
18                                     ]
19
20
21                                     ; this will work ok
22     0007      02 [                dw      2 dup (foobar)
23                                     010B R
24                                     ]
25
26
27     000B  0100 [                db      256 dup (0)
28                                     00
29                                     ]
30
31
32     010B  CB                      foobar: ret
33
34                                     ; it appears that if any calculations are required, the compiler
35                                     ; truncates after the first two bytes. The assembler reference
36                                     ; manual p. 2-15 states this is done for the small assembler but it
37                                     ; does it for both assemblers.
38
39     010C  80 51 00 00              dd      60*60*24      ; incorrect
40     0110  80 51 01 00              dd      86400         ; correct
41
42
43                                     ; Now try HIGH and MASK operators together
44
45     fooi      record  f1:1,f2:1,f3:3,f4:1,f5:1,f6:1,f7:8
46     = 0200    fc1     equ   0200h
47     = 4000    fc2     equ   4000h
48
49                                     ; immediate data should be
50     0114  25 0200                and     ax,mask f5          ; 0200h
51     0117  25 4000                and     ax,mask f2          ; 4000h
52     011A  25 0040                and     ax,high (mask f2)    ; 0040h
53     011D  25 0042                and     ax,mask f5+high(mask f2) ; 0240h
54     0120  25 4200                and     ax,(high(mask f2))+(mask f5) ; 0240h
55     0123  25 0200                and     ax,fc1              ; 0200h
56     0126  25 4000                and     ax,fc2              ; 4000h
57     0129  25 0040                and     ax,high fc2         ; 0040h
58     012C  25 0042                and     ax,fc1+high fc2     ; 0240h
59     012F  25 4200                and     ax,(high(fc2))+(fc1) ; 0240h
60     0132  25 4200                and     ax,fc1 or high fc2   ; 0240h
61
62
63                                     ; 80286 PUSH IMMEDIATE instruction bugs
64                                     ; an op code of 6A requires one byte of immediate data
65                                     ; an op code of 68 requires two bytes of immediate data
66
67     0135  6A 00                  push     0                ; 6a00 correct
68     0137  6A 7F                  push     127             ; 6a7f correct
69     0139  6A FF                  push     -1             ; 6aff correct
70     013B  68 80                  push     128            ; should be 680080
71     013D  68 0080                push     word ptr 128   ; generates correct code
72     0140  68 FF                  push     255            ; should be 6800FF
73     0142  68 00FF                push     word ptr 255   ; generates correct code

```

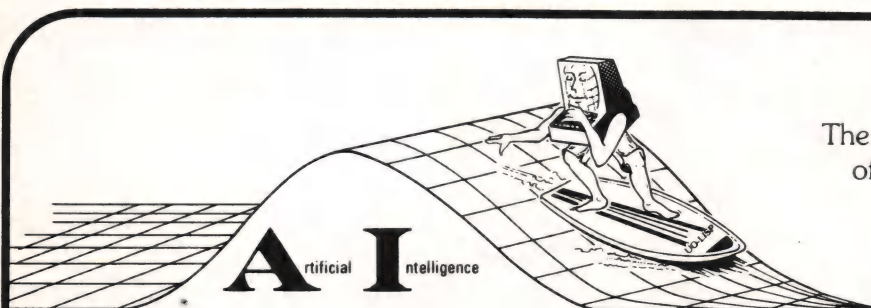


```

74      0145  6A FFFF      push  0ffffh      ; should be 6aFF or 68FFFF
75      0148  6A 34       push  byte ptr 1234h ; lost data with no error
76
77
78      ; I have to define my data before use or tell the assembler
79      ; where it is. Multi-pass compilers should be smarter
80      ; than this
81
82      014A  2E: D7       xlat  fred          ; causes (spurious) phase error
83      ; message but generates right code
84
85      014C  2E: D7       xlat  cs:fred      ; works ok
86
87      FF [             fred db 255 dup (0)
Error --- 6:Phase error between passes
88      00
89      ]
90
91
92
93      ; This bug is not very serious because the only thing wrong is
94      ; the assembly listing. The generated code is correct but the
95      ; listing of the generated code is not very neat.
96
97      024C  0D 3D [D 0A   db 13,10,13,10,61 dup (' '),13,10
98      2D
99      ]
100     0D 0A
101
102
103     028F      foos  endp
104     028F      foo  ends
105                      end      foos

```

End Listing



LISP

The preferred symbolic processing language
of the Artificial Intelligence Community

catch the next micro-wave with

UO-LISP

Not "just another pretty dialect" but the most powerful implementation of LISP available in the micro market place. For the professional engineers, researchers, and educators, UO-LISP maintains the power and flexibility inherent in LISP while providing the expected functionality of mainframe LISP systems. (+) **UO-LISP steps beyond the competition and provides a real source to native code compiler.**

CPU Family	Operating System	Production System	Learn System	Production plus Learn System
8086	MS-DOS	150 ⁰⁰	85 ⁰⁰	185 ⁰⁰
	PC-DOS	150 ⁰⁰	85 ⁰⁰	185 ⁰⁰
	CPM/86	available soon	—	—
Z80	CPM	125 ⁰⁰	85 ⁰⁰	160 ⁰⁰
	TRSDOS	80 ⁰⁰	N/A	N/A

For MORE DETAIL AND TO ORDER:
Send for *FREE* brochures and order forms.

NORTHWEST COMPUTER ALGORITHMS

P.O. Box 1747, Novato, CA 94948
(415) 897-1302

Circle no. 72 on reader service card.

ANNOUNCING! DR. DOBB'S COMPLETE TOOLBOX OF

*Dr. Dobb's Journal,
the most respected source of
technical software information available,
brings you this collection
of powerful programming tools for C.*



New, from M&T Publishing and
Brady Communications . . .

Dr. Dobb's Toolbook for C

Item #005

A comprehensive library of valuable C code.

Many of Dr. Dobb's most popular articles on C from sold-out issues are updated and reprinted in this unique reference, along with new C programming tools, **THE TOOLBOX** contain's C-compilers, line editors, assemblers, text processing programs, and more! Dr. Dobb's Journal offers **The Toolbook** in a special hardbound edition for only \$29.95. You'll find:

- J.E. Hendrix's famous *Small C Compiler v. 2* and *A New Library for Small C* (also available on disk)
- Never before published: Hendrix's new *Small-Mac: An Assembler for Small C* and *Small Tools: Programs for Text Processing* (both available on disk)
- Ron Cain's original *A Small-C Compiler for the 8080's*
- Plus many useful programming tools in C

Also from M&T Publishing and
Brady Communication—

The Small-C Handbook

Item #006 or #006A

The Small-C Handbook by James Hendrix is a valuable resource of information about the Small-C Compiler. In addition to descriptions of the language and the compiler, **The Handbook** also contains the entire source listings of the compiler and its library of arithmetic and logical routines. A perfect companion to the Hendrix Small-C compiler offered by Dr. Dobb's on disk, **The Handbook** even tells how to use the compiler to generate new versions of itself. All this is in **The Handbook** for only \$17.95, Item #006; only \$22.95 for **The Handbook** with the **MS/PC DOS Handbook Addendum**, Item #006A.

While both **The Toolbook** and **The Handbook** provide documentation for the Small-C Compiler on disk, **The Handbook** provides more complete documentation and is available with an Addendum for MS/PC DOS versions.

DR. DOBB'S C TOOLS ON DISK

To complement The Toolbook, Dr. Dobbs' also offers the following programs on disk. Source code is included, and except where indicated, both CP/M and MS/PC-DOS versions are available.

Small C Compiler Item #007

Jim Hendrix's Small-C Compiler is the most popular piece of software published in Dr. Dobbs' 10-year history. Like a home study course in compiler design, the Small-C Compiler and The Small-C Handbook provide all you need to learn how compilers are constructed, as well as teaching the C language at its most fundamental level. The Small-C Compiler is available for \$19.95 in both CP/M and MS/PC DOS versions.

Both The Toolbook and The Small-C Handbook provide documentation for the compiler; however, The Handbook provides more complete documentation for both versions, and the MS/PC DOS Small-C Handbook Addendum is recommended in addition to The Handbook for MS or PC DOS specific documentation.

Small-Mac: An Assembler for Small-C Item #012A

Small-Mac is a macro assembler designed to stress simplicity, portability, adaptability, and educational value. The package features simplified macro operators, C-language expression messages, object defined machine instruction table. Included programs are: macro assembler, linkage editor, load-and-go loader, library manager, CPU configuration utility, and dump relocatable files. This program is available with documentation in the Small Mac Manual for \$29.95. For CP/M systems only.

Small Tools: Programs for Text Processing Item #010A

This package consists of programs designed to perform specific functions on text files, including: editing, formatting, sorting, merging, listing, translating, searching, changing, concatenating, encrypting and decrypting, replacing spaces with tabs and tabs with spaces, counting characters, words, or lines, and selecting printer fonts. Source code only is included. This program is available with documentation in the Small Tools Manual for \$29.95. Both CP/M and MS/PC DOS versions available.

Special Holiday Packages — 20% Off!

Now, for almost 20% off the individual product prices, you can order a complete set of Dr. Dobbs' C programming tools for your CP/M or MS/PC DOS system!

CP/M C Package Only \$99.95 Item #005A

Ordered individually, these C tools sell for over \$120! Order the CP/M C Package now and you'll receive Dr. Dobbs' Toolbook, The Small-C Handbook, The Small-C compiler on disk, along with documentation in The Small Tools Manual, The Small-Mac Assembler on disk, and The Small-Mac Manual—all for only \$99.95!

MS/PC DOS C Package Only \$82.95 Item #005B

Individually, these tools sell for over \$100! Order now and you'll receive: Dr. Dobbs' Toolbook, The Small-C Handbook with the MS/PC DOS Addendum, The Small-C Compiler on disk, The Small Tools Text Processor on disk, along with documentation in the Small Tools Manual—all for only \$82.95!

Books	Dr. Dobb's Toolbook			\$29.95
	Small-C Handbook			\$17.95
	Small C Handbook with MS/PC DOS Addendum			\$22.95
Disks		Check Format		
		CP/M	MS/PC DOS	
				\$19.95
	Small-C Compiler (Toolbook or Handbook recommended for documentation)			\$29.95
	Small Tools Text Processor with Small Tools Manual			\$29.95
	Small Mac Assembler with Small Mac Manual (for CP/M only)			\$99.95
SPECIAL HOLIDAY PACKAGES	CP/M C Package			\$82.95
	MS/PC DOS C Package			
Add applicable sales tax for shipping in U.S., \$4.25				
Tax				Subtotal %
Shipping				TOTAL

of the following formats: ☐ 8" SS/ST

For CP/M system disks only, please specify one of the following formats:
☐ Kaypro ☐ Apple ☐ Zenith Z-100 DS/DD ☐ Osborne ☐ 8" SS/SD
 Inquire about other formats.

PAYMENT MUST ACCOMPANY YOUR ORDER.
 — Check Enclosed ☐ VISA ☐ M/C ☐ Amer. Exp.
 — Charge my

Card # _____ Exp. _____
 Signature _____ State _____ Zip _____
 Name _____
 Address _____
 (Please use street address)
 City _____

Please return this coupon along with your payment to: Dr. Dobbs' Journal, 2464 Embarcadero Way, Palo Alto, CA 94303 For credit card orders, call toll-free: 1-800-528-6050 Ext. 4001 Refer to item numbers.
 Please allow 4-8 weeks for delivery

Introduction to Using Your MacSCSI Host Adapter

by John Bass

The MacSCSI host adapter was designed to work with most SASI- and SCSI-compatible controller cards. It is also a reasonable starting point for other custom Mac interfaces. Source software written in Aztec C is provided for custom configuring your system. If your controller is Xebec S1410 compatible you should be up and running with little trouble.

Changing the Parameters for the Drive and Controller

For controllers other than the Xebec S1410, you may need to modify the last byte of the format, read, and write commands to reflect the vendor-specific requirements. The Xebec S1410 uses this byte to encode the step rate for the drive—as shipped, the code is set up to use a Seagate ST506 drive with the half-step strap enabled. If you are using a Xebec S1410 with another drive, consult your controller manual for the correct drive code to use. Then change the `cmd.sc_vendor` assignment from 1 to the correct drive-type code.

Because the Xebec defaults to an ST506-drive configuration we didn't need to issue any additional commands to define the drive configuration to the controller. This is an area that is very vendor specific—read your controller documentation very carefully. You will want to read the descriptions for the format, read, and write commands closely. Also check the error-code definitions carefully because they vary greatly between some controllers.

You may need to issue a Set Media or Set Mode command that defines the drive parameters for number of heads, cylinders, write precomp, write reduce, and so on. Some drives, such as the Xebec Owl, have an embedded controller and don't need such a setup

parameter. Some controllers encode the parameters at format time and write them on the disk for future use. Other controllers require a Mode Select or other command to set the drive parameters prior to the first format, read, or write command.

The long evolution from early SASI controllers to the current proposed SCSI specification has left a lot of nonconforming controllers around the marketplace, and many more nonconforming controllers are likely to be produced before the standard is finished and adopted. Until then, the software drivers for each host adapter will need to be customized per controller.

A copy of the current SCSI (ANSI X3T9.2) specification is available from the American National Standard publications office. For NCR-5380 information contact your local NCR Microelectronics representative or write to the NCR Microelectronics Division, 1635 Aeroplaza Dr., Colorado Springs, CO 80916; (800) 525-2252.

Mounting Drives Inside or Outside the Mac

Several 5-, 10-, and 20-megabyte 3½-inch drives are now on the market, and controllers in the same form factor are available as well. The easiest way to mount a 3½-inch drive, controller, and power supply internally is to fabricate a sheet metal and/or Plexiglass mount and screw it inside the plastic back of the Mac. To get AC for your power supply, install a power cable to the video board after the switch and AC filters. Be sure to leave a long enough service loop on both the power and SCSI cables to make getting the back off easy.

Other internal mounting configura-

tions require more attention to sheet metal design and choice of controller, drive, and power supply. Convection cooling can be enhanced by mounting everything on the right side and using the sheet metal and Plexiglass to form a chimney extending from the bottom to the vents at the top of the case. EMI fields from the power supplies, flyback, and yoke coils present some problems in getting the disk to operate without errors—proper selection of shielding material and position is critical.

The easiest way to interface drives to the MacSCSI is to mount them externally in a separate box, so that physical mounting, heat, and EMI problems are less severe. Several manufacturers have standard line enclosures. Some of these enclosures are complete with power supply, cables, fans, and maybe a drive of your choice.

External drives require an external cable. The SCSI standard specifies an EMI shielded cable, plug, and header assembly. We use AMP part number 1-499977-0 for the panel-mounted connector and AMP part number 102793-4 for the ground plane (shield assembly) that mates with the connector. Use the ground plane only if you plan to use a SCSI standard shielded cable—it will interfere with a standard nonshielded IDC ribbon-cable connector.

Installing the MacSCSI Board

Getting the back off a Mac requires a Torx 5 screwdriver with a long shaft. There are five screws: two are located at the top in holes under the handle, two are on the back toward the bottom above the connectors, and one is located behind the battery cover. Carefully use a wooden ruler or the battery cover to ease the crack along

PERFORMANCE PACKAGE

Blaise Computing Inc. introduces the PERFORMANCE PACKAGE™ for Turbo Pascal programmers.

TURBO PASCAL

Turbo ASYNCH™ With Turbo ASYNCH, you can be in constant touch with the world without ever leaving the console. Rapid transit at its best. Turbo ASYNCH is designed to let you incorporate asynchronous communication capabilities into your Turbo Pascal application programs, and it will drive any asynchronous device via the RS232 ports, like printers, plotters, modems or even other computers. Turbo ASYNCH is fast, accurate and lives up to its specs. Features include...

- ◆ Initialization of the COM ports allowing you to set all transmission options.
- ◆ Interrupt processing.
- ◆ Data transfer between circular queues and communications ports.
- ◆ Simultaneous buffered input and output to both COM ports.
- ◆ Transmission speeds up to 9600 Baud.
- ◆ Input and output queues as large as you wish.
- ◆ XON/XOFF protocol.

The underlying functions of Turbo ASYNCH are carefully crafted in assembler for efficiency, and drive the UART and programmable interrupt controller chips directly. These functions, installed as a runtime resident system, require just 3.2K bytes. The interface to the assembler routines is written in Turbo Pascal.

The Turbo Pascal PERFORMANCE PACKAGE™ is for the serious Turbo Pascal programmer who wants quality tools to develop applications. Every system comes with a comprehensive User Reference Manual, all source code and useful sample programs. They require an IBM PC or compatible, utilizing MS-DOS version 2.0 or later. There are no royalties for incorporating PERFORMANCE PACKAGE functions into your applications.

Turbo POWER TOOLS and Turbo ASYNCH sell for \$99.95 each, and they may be ordered directly from Blaise Computing, Inc. TO ORDER, call (415) 540-5441

BLAISE COMPUTING INC.

NEW!

Turbo POWER TOOLS™

Turbo POWER TOOLS is a sleek new series of procedures designed specifically to complement Turbo Pascal on IBM and compatible computers. Every component in Turbo POWER TOOLS is precision engineered to give you fluid and responsive handling, with all the options you need packed into its clean lines. High performance and full instrumentation, including...

- ◆ Extensive string handling to complement the powerful Turbo Pascal functions.
- ◆ Screen support and window management, giving you fast direct access to the screen without using BIOS calls.
- ◆ Access to BIOS and DOS services, including DOS 3.0 and the IBM AT.
- ◆ Full program control by allowing you to execute any other program from within your Turbo Pascal application.
- ◆ Interrupt service routines written entirely in Turbo Pascal. Assembly code is not required even to service hardware interrupts like the keyboard or clock.

Using Turbo POWER TOOLS, you can now "filter" the keyboard or even DOS, and create your own "sidekickable" applications.

YES, send me the Best for the Best! Enclosed is _____ for
☐ Turbo ASYNCH ☐ Turbo POWER TOOLS. (CA residents add
6 1/2% Sales Tax. All orders add \$6.00 for shipping.)

Name: _____ Phone: _____
Shipping Address: _____ State: _____ Zip: _____
City: _____ Exp. Date: _____
VISA or MC #: _____

Turbo Pascal is a trademark of Borland International. Turbo POWER TOOLS, Turbo ASYNCH and PERFORMANCE PACKAGE are trademarks of Blaise Computing Inc. IBM is a registered trademark of International Business Machines Corporation. MS-DOS is a trademark of Microsoft Corporation.

Watch us!

◆ 2034 BLAKE STREET
◆ BERKELEY, CA 94704
◆ (415) 540-5441

the front seam apart—do not use a screwdriver as a pry because the case plastic is very soft and will show pry marks.

Remove the power and floppy cables from the digital board, then gently slide it out of the sheet metal frame. Place the board on a table with the keyboard connector toward the back and the other connector near you. Lay the MacSCSI board near the ROMs with the SCSI header near you. Remove and transfer both of the Mac ROMs onto the MacSCSI board, then plug the MacSCSI board back into the ROM sockets.

Now clip the miniplug onto pin 8 of the TSM PAL located at D1 on the right edge of the board. Lay the clip flat against the board alongside the 15.667200 crystal and secure it with a piece of masking tape across the crystal and over the edge of the board. Route the wire between the switches and toroid coil and continue alongside the metal shield to the MacSCSI board. Replace the digital

board in the frame, flexing slightly to pass the MacSCSI header past the edges of the frame. Be careful not to pull the board out of the ROM sockets. Cable the SCSI port as required, reconnect the power and floppy cables, and reassemble the case.

Bringing Up the System for the First Time

Once you have the system cabled up and have made any required software changes, try formatting the drive. Watch for the drive to be selected. On some drives you can see the positioner step from track to track, and on others you may be able to feel the drive stepping. Formatting takes a while—for 5- and 10-megabyte drives it may take several minutes, and 100-megabyte drives may need a half hour. The format routine initializes the Mac file system by writing a volume header and clearing all other blocks and then rereads all blocks to check for errors.

If you have problems with Aztec C

you can add printf's to the C I/O routines and follow the progress. Once you have the format routine and I/O routines running, compile and link the driver with the I/O routines.

If you are connecting to something other than a disk you can simply include the I/O routines in your application and drive the SCSI bus directly.

Miscellaneous Comments

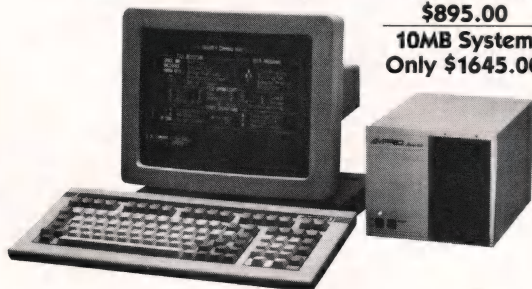
We have used MacSCSI under Finder 4.1 and Aztec C Shell 1.06D and F with a nonpartitioned 5-megabyte drive. Aztec C Shell and environment seem to run well even with a very large number of files. Finder has some limits you should keep in mind. On large volumes it gets pretty slow after 150 files. Finder also crashes sometimes when too many files get created. Deleting some files or the desktop under Aztec C sometimes makes it happy; at other times it seems that you need to back up the disk under Aztec C Shell and reformat the volume. We are working with

BOOKSHELF™

**Fast, compact, high quality,
easy-to-use CP/M system**

Series 100

**Priced from
\$895.00
10MB System
Only \$1645.00**



Features

- Ready-to-use professional CP/M computer system
- Works with any RS232C ASCII terminal (not included)
- Network available
- Compact 7.3 x 6.5 x 10.5 inches, 12.5 pounds, all-metal construction
- Powerful and Versatile:
 - Based on Little Board/PLUS single-board computer
 - Two RS232 serial ports
 - One Centronics printer port
 - One or two 400 or 800 KB floppy drives
 - 10-MB internal hard disk drive option
- Comprehensive Software Included:
 - Enhanced CP/M operating system with ZCPR3
 - Word processing, spreadsheet, relational database, spelling checker, and data encrypt/decrypt (T/MAKER III)
 - Operator-friendly shells; Menu, Friendly™
 - Read/write and format dozens of floppy formats (IBM PC-DOS, KAYPRO, OSBORNE, MORROW...)
 - Menu-based system customization
- Expandable:
 - Floppy expansion to four drives
 - Hard disk expansion to 60 megabytes
 - SCSI/PLUS™ multi-master I/O expansion bus

AMPRO
COMPUTERS, INCORPORATED

67 East Evelyn Ave. • Mountain View, CA 94041 • (415) 962-0930 • TELEX 4940309

T/MAKER III is a trademark of T/Maker Company.
IBM is a registered trademark of International Business Machines.
Z80A is a registered trademark of Zilog, Inc.
CP/M is a registered trademark of Digital Research.

Circle no. 158 on reader service card.

Release 2.0

C-INDEX™

Variable Length Record Management For C

C-INDEX is a state-of-the-art data management function library for C programmers. Ideal for all data and text based applications. No other package can give you the performance, capability, and portability of C-INDEX. To make sure you are a satisfied customer, we offer a 30 day money-back guarantee. Ask us about it.

- variable length data storage with B+Tree indexing
- high performance, easy to use
- large and small models, fully transportable source
- IBM PC format: Lattice, Microsoft, C86, others
- Macintosh format: Consulair, Manx, others

C-INDEX/FILE (\$99) Object code package.

C-INDEX/PRO (\$195) Partial source code, no royalties.

C-INDEX/PLUS (\$395) Complete transportable source code, no royalties.

Trio Systems

2210 Wilshire Blvd., Suite 289
Santa Monica, CA 90403
213/394-0796

Circle no. 150 on reader service card.

Apple to find the solutions to these problems.

We have used the driver with and without a cache with some interesting results. For small cache sizes the overhead of maintaining the cache slows the overall operation down. For the Aztec C environment the minimum size is roughly the sum of all the programs, libraries, and include files in use—it is easier to start up the RAM disk and copy everything to it. For large cache sizes it is a pain to run large programs or the switcher because you need to restart the driver with a smaller cache size. The overall conclusion for our working environment is that, although the cache can create RAM-disk speeds, it sometimes creates floppy speeds. The MacSCSI hard disk is nearly as fast as the RAM disk anyway, and maintaining the cache just becomes a headache when switching environments.

Another minor annoyance is that you can't remount the MacSCSI volume if you eject it from an open box—never do so. We are working on adding the code to trap the ejects and ignore them. The eject solution, multiple volumes via partitions, and other improvements will be available in our 2.0 software release within a few weeks.

We are collecting changes for other controller-and-drive combinations to be available on our update disk. If you would like to share your changes, send us a disk with your modifications and we will return it with the current updates. Mail your changes to Fastime, P.O. Box 12508, San Luis Obispo, CA 93406.

DDJ

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 198.

Thinking of the C Language?

THINK COMPUTER INNOVATIONS

NEW!!

C86 VERSION 2.3 with Source Level Debugging Support

The C language has rapidly become the development language of choice for applications ranging from Operating Systems to Accounting Packages. WHY? Its structured approach and extreme portability make it perfectly suited to today's fast-paced environment.

Of all of the C Compilers available for PC/MSDOS, more programmers choose COMPUTER INNOVATIONS' C86. WHY? Because it's part of a COMPREHENSIVE family of C products with an unparalleled reputation for performance, reliability, and stability.

C86 2.3 C COMPILER

C for PC/MSDOS began with C86 and today it remains perhaps the most solid, stable C Compiler available. Even competitor's ads show C86 as a consistent top level performer in benchmark testing.

Version 2.3 offers a host of new features including source level debugging support and a 40% boost in compilation speed. Call for complete specifications.

**COST: \$395 UPDATE TO 2.3: \$35 w/old diskettes NOT COPY PROTECTED
CALL ABOUT VOLUME DISCOUNTS**

LEARN C INTERACTIVELY WITH INTRODUCING C

Intimidated by rumors about the difficulty of learning C? Need to train your staff quickly? INTRODUCING C can help. INTRODUCING C combines a thorough, self-paced manual with a unique C interpreter to provide a fast, efficient method of learning C. Designed for both professional and casual programmers, it provides a comprehensive understanding of important C concepts such as standard K&R syntax and operators, full structures and unions, arrays, pointers, and data types. Requires IBM PC, XT, or AT with one disk drive and 192K bytes of memory.

COST: \$125 - NOT COPY PROTECTED

CI PROBE SOURCE DEBUGGER

Take advantage of C86 2.3 source level debugging support with CI PROBE. Cut down program development time and save money! CI PROBE is highly economical yet has the features of debuggers costing far more.

COST: \$225 - NOT COPY PROTECTED

C-TERP C86 COMPATIBLE INTERPRETER

The C-TERP INTERPRETER is a full K&R implementation that allows you to write code and execute it immediately without the compile and link steps. Once you have your program running with C-TERP you can compile the code (without alterations) with C86 for fast, efficient executable files. C-TERP requires 256K, 512K is recommended.

**COST: C86 version - List Price: \$300, Special Computer Innovations Price \$250.
Combined C86 & Lattice version - List Price: \$400, Special Computer Innovations Price \$350.**

Start With Us, Stay With Us

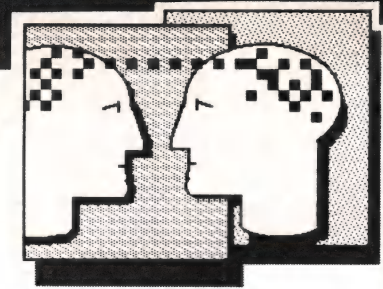
Computer Innovations offers a complete range of products that let you enter the C environment and create applications with the most advanced set of development tools available. Unparalleled tech support assures that you're always at the height of productivity.

To order call: **800-921-0169**

 **COMPUTER INNOVATIONS, INC.**

980 Shrewsbury Ave., Tinton Falls, NJ 07724 • (201) 542-5920

C-TERP is a trademark of Gimble Software. Prices and specifications subject to change without notice.



by W. E. Wilson

Software running on a Z80-based system under CP/M-80 will frequently perform better than comparable software running on an IBM PC under MS DOS. For example, Microsoft BASIC-86 is actually slower on the IBM PC than BASIC-80 on a Z80-based system. Programs running under CP/M-80 also are frequently smaller than their counterparts on MS DOS. Translate a Z80 assembly-language program into 8086 assembly language and watch the code expand.

Sixteen-bit systems do have a number of advantages over their 8-bit predecessors. You can address more memory and can have better, even color, graphics. The PC can communicate easily with mainframe systems and will probably be more compatible with future developments than the older 8-bit systems. On the other hand, there are still many more software packages available for CP/M than there are for MS DOS, some quite good and very reasonably priced.

When comparing Z80-based systems to the IBM PC, there are some basic facts that you should keep in mind. First of all, the limited addressable memory space of 8-bit systems used to force programmers to write tight, rapidly executing code. With the advent of 16-bit systems and more addressable memory space, software developers were no longer compelled to keep code size small, and they shifted to less efficient programming techniques, including writing in system-independent, high-level languages. Second, most popular 16-bit programs were originally written for 8-bit systems. Developers

have often moved these programs to the PC by translating the 8-bit code line by line into the equivalent 16-bit code. The result is that much of the potential efficiency of the 16-bit systems has been lost. For example, many of the spreadsheets that run without any problems on a 64K 8-bit system require more than 128K on a 16-bit system. Third, the memory overhead for the MS DOS operating system is so much greater than the overhead for CP/M-80 that a 128K PC is approximately equivalent to a 64K CP/M-80 system. Finally, the 5¼-inch floppy disk drives on a PC are as slow as molasses in January when compared to the typical 8-inch drives on the older CP/M-80 systems.

Mini System Software for a Micro

Back in the days before the invention of the processor on a chip and the development of modern micro systems, memory was very expensive. Mini-computer companies such as Digital Equipment Company provided overlaying linkers for use with their computers. In the early '70's the institution where I worked had a DEC PDP-15 with 64K of memory. DEC provided a software package called Chain and Execute that allowed you to run a very large program on the PDP-15 by segmenting it into a series of overlays. I used Chain and Execute to run some huge programs written in FORTRAN for the IBM 360 on the little DEC system. Not very much is written on the subject of overlays¹, but this technique is used extensively in mainframe systems and in multiuser operating systems.

The Linking Process

The creation of an executable program from a number of source files

involves several steps. First, the source files are individually read by the compiler and converted into one or more object modules. Some compilers produce relocatable object code directly. Small-C and its derivatives generate assembly-language output that must be assembled. The following discussion applies to compilers of the latter type only.

It is not practical to have the compiler generate all the code needed for each module. Instead, code is generated only for operations specific to the particular module. General-purpose functions and subroutines from the run-time library are merely referenced by the compiler and only later linked into the program. These external references are "place holders" into which absolute addresses are eventually placed. For functions and variables defined within the module, the compiler generates references relative to the beginning of the module. This means that if the module is eventually loaded at address *L*, then all references will be correct if *L* is added to them.

An assembler then reads each individual module created by the compiler and generates for each one a relocatable object module. This module is called relocatable because it can be relocated anywhere in memory.

The final stage is linking. The linker searches the run-time library for any referenced subroutines and functions. It loads these subroutines and functions along with the relocatable object modules into their final positions in contiguous blocks of memory. Finally, it fills in the absolute addresses for externals and adjusts the relative addresses for internals.

A standard linker builds programs in memory. Thus, the maximum program size is equal to the total memo-

W. E. Wilson, Washington State University, Nuclear Radiation Center, Pullman, WA 99164

ry space minus the space used by the operating system and the linker. An overlaying linking loader links the program on disk rather than in memory and thus can link a program that will fill the entire available memory space or more. There is, however a penalty paid in speed. An overlaying linking loader must read every file twice—once to locate the globals and determine the size of the module and once to do the actual linking.

ZLKO

ZLKO from ZEE MICROWARE is an overlaying linking loader for Z80-based systems. The most important feature of ZLKO comes into play when a program is too big to fit into available memory space. The standard solution to this problem is to use the call-next-program technique. This technique involves dividing the program into a selected number of segments, each small enough to fit individually into memory. Each sequential segment except the last must load and execute the next segment. This method is only supported by a few compilers and has a number of disadvantages:

- It is not possible to return to a calling segment
- All segment intercommunication must be via disk files
- Selective segment execution is not possible

All of these weaknesses can be overcome by using the overlaying capabilities of ZLKO.

The user segments the program into a small root (main) module and one or more overlays, each of which is composed of a number of subroutines and functions. The subroutines are grouped into overlays according to criteria selected by the user. ZLKO then constructs a multilevel "tree" of overlays in which the overlays at any given level can call overlays at the next lower level. The execution of the segmented program begins when the root module is executed. The root will then activate other overlays by calling the subroutine "ovrlay" that has been linked with the root segment. The root segment will stay in memory



RUN. EDIT. RUN. EDIT. RUN. COMPILE.

*If you're not using a C interpreter,
you're not getting the most out of your C compiler.*

Stop cursing at your compiler for slow compilation times. Don't blame your compiler. It was designed to produce great executable code. Try *RUN/C Lite* instead. *RUN/C Lite* was designed to let you edit, test, and debug program modules interactively, without delay. *RUN/C Lite* and your compiler are a natural combination for fast C program development.

NOW THERE ARE TWO RUN/C'S. ANNOUNCING RUN/C LITE AND RUN/C PROFESSIONAL™!

Thousands of RUN/C™ users were happy with RUN/C just the way it was. But some people (mainly professional programmers) wanted more. So we developed two RUN/C packages. *RUN/C Lite* provides all the standard RUN/C advantages. *RUN/C Professional* adds new and powerful facilities for serious software development.

Now with *RUN/C Professional*, you can dynamically load and unload previously compiled functions, and execute these functions in real-time at compiled speed. Use your favorite libraries (Greenleaf, C-Food Smorgasbord, Multi-Halo, GSS™ Graphics, etc.) as part of your RUN/C interpreted code. No need for special configuration, as with some interpreters. (To create new Loadable Libraries, Lattice™ large-model is required.)

DEBUGGING TOOLS

Besides the standard TRON and PRON, *RUN/C Professional* includes an array of new source-code debugging facilities, to test your modules before compiling. TRACE by line ranges. Immediate mode allows you to execute by function. Set multiple breakpoints. Read and/or change the value of a variable and continue with the execution of a program.

FULL-SCREEN EDITOR

A full-screen editor with Wordstar™-like commands is built into RUN/C for program editing convenience.

For immediate delivery or more information, contact:

LIFEBOAT ASSOCIATES

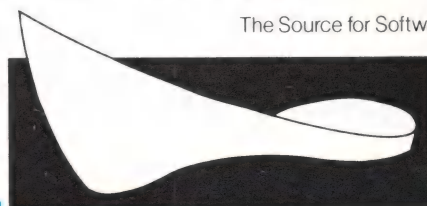
1651 Third Avenue
New York, NY 10128
Telex: 424490 (LBSOFT UI)
Telephone: 800-847-7078
212-860-0300 (in New York State)

Dealers Welcome Visa, Mastercard, American Express accepted

INTERNATIONAL SALES OFFICES

Italy: Lifeboat Associates, S.p.A. Milan Phone (02) 656 841
Japan: Lifeboat Japan, Tokyo Phone (03) 293 4711
Scandinavia: Lifeboat Scandinavia, Malmö, Sweden
Phone (46) 40 10035
England: Grey Matter Ltd, Asburton, Devon Phone (44) 364 53499
Australia: Fagari Microprocessor Associates, Middle Park, Victoria
Phone (61) 3699 9899

Lifeboat, The Source for Software with Full Support, and Lifeboat logo are trademarks of Lifeboat Associates. RUN/C, RUN/C Lite, RUN/C Professional, and Loadable Libraries are trademarks of Age of Reason Co. Lattice is a registered trademark of and C Food Smorgasbord is a trademark of Lattice, Inc. Wordstar is a trademark of MicroPro International Corporation, Inc. GSS is a registered trademark of Graphic Software Systems, Inc. Multi-Halo is a trademark of Media Cybernetics. Greenleaf is a trademark of Greenleaf Software.



The Source for Software with Full Support™

LIFEBOAT™

© 1985 Lifeboat Associates
ALL RIGHTS RESERVED

Circle no. 125 on reader service card.

0100-0102 JP ENTRY
 0103-0107 Identification Code
 0108-0109 End + 1 of Last Overlay or first free memory location above overlay system
 010A-010B End + 1 of Root Segment or the start of first overlay
 010C-0114 Eight Character Root File Name

Byte 0-1 Overlay Transfer Address (Load Address)
 Byte 2-3 End + 1 of Segment
 Byte 4-11 Eight Character Segment File Name

Figure 1
Root and Overlay Segment Headers

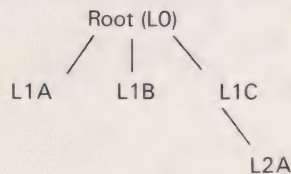


Figure 2
Structure of the test overlay system

```

ROOT(MAIN);OVINIT,OVRLAY,LO;      END;
END;                               OVL(1,L1C);
SYMFIL(CTEST);                     INCL(L1C);
OVL(1,L1A);                         END;
INCL(L1A);                         OVL(2,L2A);
END;                               INCL(L2A);
OVL(1,L1B);                       END;
INCL(L1B);                         DONE;
  
```

Figure 3
ZLKO Command File Used to Create Overlay System

ZLKO Command Summary

CMND	= Take Command Input From Specified File
DCOM	= Display Common Block Information
DMAP	= Display Global and External Symbols
DONE	= Complete Current Segment and Quit
DOVL	= Display Overlay Structure
DSEG	= Display Information On Current Overlay Segment
DUND	= Display Undefined Symbols
END	= Complete Current Segment, forces a SRCH
HELP	= Display Command List
INCL	= Include The Listed REL Files in The Current Segment
OVL	= Build Overlay (.OVL) File
PBSE	= Set Program Code Base Location
QUIT	= Quit and Return to CP/M
RELDMP	= Dump Relocatable (.REL) File
ROOT	= Build ROOT (.COM) File
SRCH	= Search Specified (.REL) Library
STOP	= Quit Building Current Segment and Save The Link Table
SYMFIL	= Generate Symbol File For Symbolic Debugger
?SYM	= Suppress Symbols Starting with ?
TCSW	= Turn Off Module Counting

Figure 4
ZLKO Help Command Output

the entire time. The overlays will be loaded into memory and control passed to each as program execution proceeds. When an overlay returns control to the higher level that activated it, the higher level may then activate another overlay beneath itself or pass control down one level. Each overlay on the same level in the tree is loaded into memory starting at the same location.

Routines in an overlay can reference any symbol defined within that overlay and all global symbols located in any active higher-level overlay up to and including the root segment. Global symbols and routines in the root segment may be referenced from any segment. Frequently needed routines and functions should be included in the root segment. Routines in an overlay cannot, however, reference symbols or routines defined in an overlay on the same level or in a segment lower down the tree.

ZLKO Usage

ZLKO places a 21-byte header on the root-segment file and a 12-byte header on each overlay-segment file (see Figure 1, page 116). The headers include the primary CP/M file name of the segment as well as address information. In the case of an overlay segment, the overlay name and transfer address are used by the overlay subroutine to load the overlay. The execution process starts at the root segment and proceeds by descending down and climbing back up each branch of the overlay tree structure one branch at a time.

Using ZLKO with C offers one significant advantage over using it with a language like FORTRAN. FORTRAN employs the call-by-reference technique. Hence, arguments cannot be passed to a called overlay except by means of the COMMON block in the root segment. On the other hand, C uses the call-by-value technique. Thus, values can easily be passed to a function in an overlay. C passes the arguments by pushing them on the stack, and the function in the called overlay simply pops them off the stack.

ZLKO does not automatically insert code into the root segment to set the

stack pointer because the required call to `main()` in the first routine in the root segment will take care of setting the stack pointer. That is, this operation is handled by the C compiler, which expands the call to `main` to include setting the stack pointer.

Most Small-C compilers use the symbol `__memptr` to designate the first free memory location above the program code space. The Q/C compiler I used to test ZLKO uses the symbol `$MEMORY` for compatibility with the L80 linking loader. Under ZLKO, the value to be used for `$MEMORY` or `__memptr` is stored in the 8th and 9th bytes of the root-segment header. This value must be extracted from the header and stored in location `$MEMORY` or `__memptr` in the compiled C program before program execution commences.

This operation is readily accomplished by using the `ovinit` function shown in Listing One (page 119). `Ovinit` must be linked into the root segment and must be the first function called from within `main()`. The overlay function supplied with ZLKO must be linked into the root segment following `ovinit`. Each time an overlay is to be loaded, the user program must make a call to the overlay function using a pointer to the overlay name as the last argument of the overlay function call. The overlay test (see next section) demonstrates the use of the overlay function. Overlay will load the specified overlay at the transfer address specified in the header and then execute the overlay.

ZLKO Test

I tested the ZLKO overlaying linking loader with a simple overlay system written in C. The test program was compiled with the Q/C compiler and assembled with the CWA assembler distributed by The Code Works. The structure of the test overlay system is shown in Figure 2 (page 116) and the command file used to create the overlay system in Figure 3 (page 116). The code for the root segment and each overlay is presented in Listing Two (page 119). Listing Three (page 125) shows the output that ZLKO sent to the console during the test. The linking process took 126 seconds

and the resulting overlay system executes in 11 seconds.

Using ZLKO

ZLKO is relatively easy to use once you understand how to pass to overlay a pointer to the overlay to be loaded by the root segment. There are, however, a couple of things you should keep in mind. First of all, because the input information required by ZLKO is extensive, the potential for keyboard errors is quite high. This problem can be solved by using a com-

mand file instead of direct input. For example, entering the command:

```
ZLKO CMND(CTEST);
```

causes ZLKO to take its input from the command file `CTEST.ZLK`. Second, you may have trouble getting accustomed to the fact that all commands require a combination of semicolon and carriage return as a terminator. Finally, although the manual adequately describes the proper use of ZLKO, it will not make

Lattice® Products Are Your Best Investment

Lattice Screen Editor \$125.00

(LSE)—Designed as a programmer's editor. LSE is fast, flexible and easy to learn. LSE's multi-window environment provides all the editor functions you need including block moves, pattern searches, and "cut and paste." LSE also offers special features for programmers such as its error tracking mode and three assembly language input modes. You can create macros or customize keystrokes, menu, and prompts to your style and preferences.

Lattice TopView Toolbasket \$250.00

—Provides more than seventy C functions for creating programs to take advantage of the IBM TopView environment. Speed TopView-specific program development with window, cursor, pointer, and printer control functions. Expedite program development with the debugging utilities included.

LMK™—Automated \$195.00

product generation utility for MS-DOS, similar to UNIX "Make."

C-Sprite™—Symbolic \$175.00

debugger for programs written in Lattice C or assembly language.

C Compilers—MS-DOS \$500.00 and PC-DOS—The industry standard.

Panel™—High performance screen design and screen management system. Available for MS-DOS, PC-DOS, and UNIX systems.

DBC II™ or DBC III™ \$250.00

C function libraries supporting indexed files that are compatible with dBASE II® or dBASE III.™

C-Food Smorgasbord™ \$150.00

Includes a BDC decimal arithmetic package, I/O functions, IBM-PC BIOS interface, terminal independence package, plus many other utility functions.

With Lattice products you get *Lattice Service* including telephone support, free updates during the warranty period, notice of new products and enhancements when you register, and a money-back guarantee.

Call us today for further information on these, or our many other C programs, utilities, and tools! Corporate site license agreements are also available.



Lattice, Inc.
P.O. Box 3072
Glen Ellyn, IL 60138
Phone (312) 858-7950
TWX 910-291-2190

International Sales Offices

Benelux: De Vooght. Phone (32)-2-720.91.28.
England: Roundhill. Phone: (0672) 54675
Japan: Lifeboat Inc. Phone: (03) 293-4711

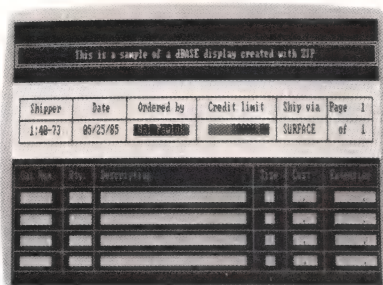
dBASE & PASCAL HAVE NEVER LOOKED BETTER.

ZIP™ gives you friendlier, more functional dBASE II/III and Turbo Pascal screens and printed reports quickly and easily.

With ZIP, you just draw your lines and boxes, set colors, type in your prompts and place your fields and variables where you want them, then ZIP writes a routine that is ready to run.

Contact Magnum Data, Inc., 627 S. Plymouth, Los Angeles, CA 90005. (213) 937-0848. And add ZIP to your programs today.

■ Writes fully-accessible source code for an unlimited number of 24 x 80 screens and printed reports up to 88 x 132 in seconds, saves you hours.



■ Supports PICTURE, COLOR and CLEAR in dBASE and Pascal.

■ Supports Pascal character, string, integer, real, Boolean data types.

■ Supports all standard printers:

translates graphics to printable characters.

■ DOS 2.0 or higher on IBM PC/XT/AT, Compaq, Corona.

■ \$50 ea., \$75 for both (US funds). Discounts start at 5 units, licensing available. \$3 S & H, plus tax in California.

Call Operator 199 at 1-800-437-5200 (in Maryland, 1-800-638-8890)



Magnum
DATA, INC.

dBASE II & III™ Ashton-Tate. Turbo Pascal™ Borland International.
© Magnum Data Inc. 1985

Circle no. 69 on reader service card.

Scrap your LINKER

with

FASTER C

Reliably:

CUT Compile times (by 15% to 55%)

CUT Testing times (by 12% to 37%)

HOW: FASTER C keeps the Lattice C or C86 library and any other functions you choose in memory. It manages a jump table to replace the LINKER and immediately execute your functions. You can also CALL active functions interactively to speed your program debugging. It includes many options for configuration and control.

"Automatic" support for new libraries by reading the .OBJ files makes support for new libraries quick and simple.

AVAILABLE FOR PC-DOS, IBM-AT,
AND ANY 256K MSDOS SYSTEM.

ONLY \$95.

Full Refund if not satisfied

during first 30 days.

Call 800-821-2492

**Solution
Systems™**

335- D Washington St., Norwell, Mass. 02061
617-659-1571

Circle no. 147 on reader service card.

you an expert at using overlays. All of the commands accepted by ZLKO are shown in Figure 4 (page 116) in the form of a printed copy of the output of the ZLKO Help command. ZLKO is quite verbose and lets you know what is going on at all times. Part of the verbosity is because of the fact that ZLKO displays each component of a command string as it is being parsed.

Conclusion

ZLKO is an easy-to-use, relatively efficient overlaying linking loader that is particularly useful to anyone using a Small-C compiler. ZLKO will allow the user to run very large programs under CP/M-80 and to construct fancy software in which a menu-selection operation determines which overlay (program) is executed next. There are at least two other overlaying linkers available for use with CP/M-80, but they are much more expensive than ZLKO and only provide a few additional capabilities. ZLKO only supports a multiple-file overlay file system, whereas some others also support a single-file overlay file system that uses the random-access capabilities of CP/M-80. A single-file overlay system will execute slightly faster but is more difficult to construct and debug.

Do not discard your older 8-bit system if you can perform the tasks you need to accomplish by simply purchasing an overlaying linking loader such as ZLKO. ZLKO is available from Elliam Associates, 2400 Bessemer St., Woodland Hills, CA 91367.

Notes

¹ A good discussion of overlays is provided by S. H. Kaisler, *The Design of Operating Systems for Small Computer Systems*, John Wiley, 1983.

DDJ

(Listings begin at right)

Reader Ballot

Vote for your favorite feature/article.
Circle Reader Service No. 199.

Listing One

```
/* Get address of first free memory location from ROOT header
and put it in location $MEMRY. This version is for use with the
Q/C compiler. Change $MEMRY to _memptr for use with small-c */
```

```
ovinit()
{
# asm
    EXT    $MEMRY
    LD      HL, (0108H)    ;Get 1st free mmry addr
    LD      ($MEMRY), HL   ;Put in $MEMRY
# endasm
}
```

End Listing One

Listing Two

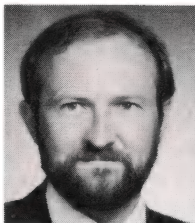
```
/* Root program L0.C for testing ZLKO with Q/C */

#include <qstdio.h>

main()
{
char *ovla, *ovlb, *ovlc;
ovla = "LlA";
ovlb = "LlB";
ovlc = "LlC";
/* initialize Root Segment For Overlays */
ovinit();
printf("ZLKO Overlaying Linker Test with Q/C\n\n");
printf("Starting at level zero\n");
printf("Call Overlay LlA\n");
```

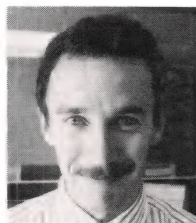
(Continued on next page)

ConcurrentTM PC DOS Works For These People



"PhoneXpressTM, our voice messaging product, needs the powerful multi-tasking environment that Concurrent PC DOS offers. Also, our programmers love it, they can be compiling in one window and switch into an editor which increases productivity."

Dennis King
Executive Vice
President, Founder
Applied Voice
Technology



"The file manager is great. It's a step above the old DOS system. You don't have to remember commands and you don't have to read the manual."

John Martinson
Engineer
Pacific Gas
Transmission



"Concurrent PC DOS gives us the capability to attach additional terminals for a more productive office. It has a true multi-user capacity. We can now meet deadlines."

Richard Vananda
Principal
Patrick Sullivan
Associates,
Architects



"I couldn't live without the product. Time is money and the ability to switch consoles and run more than one program has been a great time saver."

Joe Capp
Project Engineer
Union Carbide



"We have created a menu driven system that is totally transparent to our customers. By utilizing the menu system and 4 consoles concurrently we are able to fully utilize Concurrent PC DOS"

Subbash Chaddha
MIS Director
Secoin Inc.

Make It Work For You!

Call (800) 443-4200 for the dealer or sales representative nearest you.

Concurrent PC DOS is a trademark and Digital Research is a registered trademark of Digital Research Inc. PhoneXpress is a trademark of Applied Voice Technology.

**DIGITAL
RESEARCH[®]**
We make computers work.[™]

Circle no. 163 on reader service card.

Listing Two

```
/* Load Overlay L1A */
    overlay(&ovla);          /* Pass pointer to ovlname */
    printf("Returned to Level zero from Level one pgm L1A\n");
    printf("Next Call Overlay L1B\n");
/* Load Overlay L1B */
    overlay(&ovlb);          /* Pass pointer to ovlname */
    printf("Returned to Level zero from Level one pgm L1B\n");
    printf("Next Call Overlay L1C\n");
/* Load Overlay L1C */
    overlay(&ovlc);          /* Pass pointer to ovlname */
    printf("Returned to level zero, Overlay Test Done\n");
}
```

```
-----
/* overlay L1A */
levella()
{
    printf("At Level one, Overlay L1A\n");
}
```

```
-----
/* overlay L1B */
levellb()
{
    printf("At Level one, Overlay L1B\n");
}
```

```
-----
/* overlay L1C */
levellc()
{
```

(Continued on page 123)

You Read **Dr.Dobb's Journal** And You Don't Subscribe?!

Save over \$23.00 off newsstand prices for 2 yrs.

Save over \$10.00 for 1 yr.

Can you afford to miss an issue with information vital to your interests? As a subscriber you can look forward to articles on Small-C, FORTH, CP/M, S-100, Compiler optimization, Concurrent Programming and more, delivered right to your door. And you'll never miss the issue that covers your project.

Yes! Sign me up for

____ 2 yrs. \$47 ____ 1 yr. \$25

____ I enclose a check/money order
____ Charge my Visa, MasterCard,
American Express
____ Please bill me later

Name _____

Address _____

_____ Zip _____

Credit Card _____ Exp. date _____

Account No. _____

Signature _____

Please allow 6-9 weeks for delivery.

3048



THE International UNIX* Event of the Year!

February 4-7, 1986 / Anaheim Convention Center / Anaheim, California

UniForum 1986, the International Conference of UNIX Users, has established itself as THE premier UNIX conference/trade show.

- Over 200 major vendors exhibiting their newest UNIX-based hardware, software, systems, services and peripherals.

- A complete tutorial and conference program. Multiple sessions target the connectivity of UNIX between the user and technical environments; the hardware and software technology for office systems and workstations; the interface between man and machine, machine and machine, and much more. Day-long tutorials provide intensive, focused material on specific subjects in UNIX... while the conference sessions highlight the latest developments in the continuing evolution of UNIX.

*UNIX is a trademark of AT&T Bell Laboratories.

- **FREE** UNIX Introductory Workshops.

- Birds-of-a-Feather impromptu sessions.

Call us NOW for your **FREE** Show-Only Badge *and* complete registration information on the dynamic conference/tutorial program.

The power and potential of UNIX are important to you. UniForum 1986 is THE computer event you can't afford to miss!



For Complete Information, Call:

800-323-5155

(In Illinois, Call: 312-299-3131)

Or Write:

UniForum
2400 East Devon Avenue
Suite 205
Des Plaines, IL 60018

Sponsored by



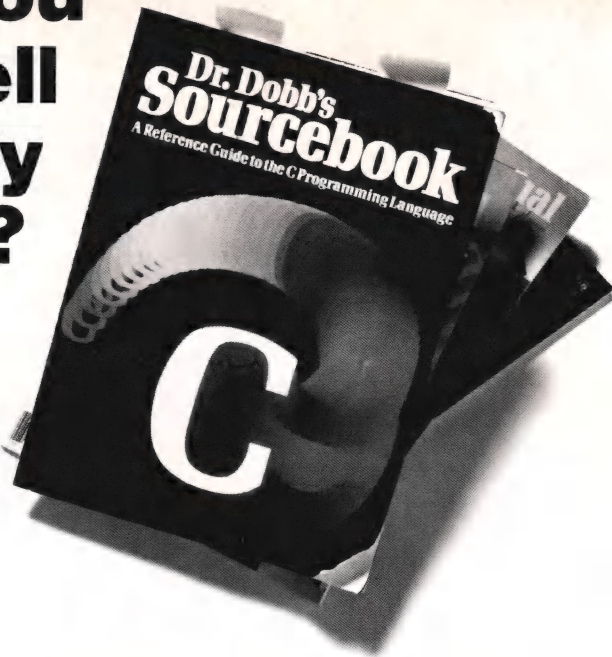
usnug

The International Network of UNIX Users

Circle **no. 120** on reader service card.

Who Says You Can't Tell A Book By Its Cover?

Dr. Dobb's Sourcebook: A Reference Guide for the C Programming Language



For years, serious programmers have relied on Dr. Dobb's Journal for the technical tools of their trade. Now, Dr. Dobb's presents the definitive programmers guide to the who, what, where, when and why of C, the leading language among software developers. This comprehensive guide to new information, products and services specific to C will be your most often-used reference!

In this valuable guide you'll find:

- An extensive directory of hardware and software services—including classes and seminars, C programming opportunities, and on-line services
- A bibliography with over 300 listings of available articles and books on C
- A comprehensive C product listing—including C compilers, graphics modules, utilities, editors and development systems, and more!
- And much more practical C programming information

At only \$7.95, no C programmer can afford to be without this unique reference.

To order by credit card, call toll free: 1-800-528-6050 ext.4001 . Ask for item 004 or mail this coupon, along with payment to **Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303**

PAYMENT MUST ACCOMPANY YOUR ORDER

_____ I enclose check/money order
_____ Please charge my _____ VISA _____ M/C _____ American Express
Card # _____ Exp. Date _____
Signature _____
Name _____
Address _____
City _____ State _____ Zip _____

Please allow 6 to 12 weeks for delivery

Please send me _____ copies of **Dr. Dobb's Sourcebook**

at \$7.95 each = _____

+ Shipping & Handling = _____

(Must be included with order. Please add \$1.50 per book in U.S. \$3.25 each surface mail outside U.S. Foreign airmail rates available on request.)

TOTAL = _____

Listing Two

```

char *ovld;
int value1, value2, sum;
ovld = "L2A";
printf("At Level one, Overlay L1C\n");
printf("Call Level two, Overlay L2A 5 times\n");
/* Load Overlay 2A */
value1 = 0, value2 = 9;
while (++value1 < 6)
{
    ++value2;
    printf("At Level one Calling L2A, value1 = %d, value2 = %d\n", value1, value2);
    /* The pointer to the overlay name must be the last parameter */
    /* The ZLKO "ovrlay" routine uses the value in the HL register- */
    /* -to find the overlay name but does not alter the stack */
    /* The called routine must have a dummy last parameter */
    ovrlay(value1, value2, &ovld);
    /* Pass values and pointer to ovlname */
}
int sum;
retval(sum)
{
    printf("Back at Level one, Returned sum value1 + value2 = %d\n", sum);
}

-----

/* overlay L2A */
level2a(arg1, arg2, arg3)
/* arg1 is value1, arg2 is value2, and arg3 is ovlname pointer */
/* dummy arg3 is required to match ovrlay function call in L1C */
int arg1, arg2, arg3;
{
    printf("At Level two, Overlay L2A\n");
    printf("Received parameter value1 = %d, value2 = %d\n", arg1, arg2);
    /* Pass sum arg1 + arg2 back to level one */
    retval(arg1 + arg2);
}
    
```

End Listing Two

(Listing Three begins on page 125)

PIK'r

Convert Mainframe Reports to LOTUS Formats Without Re-keying

- The data editor that links mainframe reports to micro spreadsheets.
- Move selected report data directly to the spreadsheet; reformat, edit, rearrange data—no re-keying! Text and numbers!
- Accepts any ASCII file, supports all major spreadsheets, databases, word processors (Lotus, dBASE, Multiplan, Visicalc)

and introducing

MACROPIK'r

Combines All the Benefits of PIK'r Plus:

- A macro facility—save and replay entire sessions with a single keystroke.
- An unlimited page size—works with reports of any size.

Both PIK'r and MacroPIK'r offer mouse support, work with IBM-PC and compatibles, any mainframe. Customer HOTLINE, tutorials and user manual included.

PIK'r—\$95

MacroPIK'r—\$249

SAMKHYA CORPORATION

47 Sixth St., Suite 3000
P.O. Box 142
Petaluma, CA 94953
(707) 763-2800

Toll Free

(800) 442-0012 US
(800) 442-5544 CA

Circle no. 171 on reader service card.

COHERENT for the AT

**At last —
a UNIX compatible O/S with elaborate
hard disk recovery routines!**

- ▶ COHERENT is the best UNIX compatible O/S for networking, process control and multiuser applications with highly efficient code generation, configurable kernel and flexible disk partitioning.
- ▶ Support available for high resolution graphics, image capture, mouse and other hardware.
- ▶ A complete COHERENT development environment that includes a screen editor and source code control system.
- ▶ Total object code compatibility between the IBM AT, XT, and PC.
- ▶ Available with UNIX System V compatible support for interprocess communication enhancements and high speed multiline communication cards (up to 38.4 Kbaud).
- ▶ Can be easily programmed into a turnkey environment. Flexible turnkey and runtime licensing available.
- ▶ Professional technical support provided.

To discuss your system requirements call: **(604) 294-6201**

INETCO Systems Ltd.

2457 Beta Ave., Burnaby, B.C., Canada V5C 5N1

COHERENT is a trademark of Mark Williams Co. UNIX is a trademark of Bell Laboratories. IBM AT, XT and PC are trademarks of International Business Machines Corp.

Circle no. 166 on reader service card.

CONTINUING THE TRADITION

DR. DOBB'S JOURNAL ANNOUNCES THE RELEASE OF BOUND VOLUME 8

Every 1983 issue together in one source

BOUND VOLUME 8 (Item #020)

DDJ turns pro. Some of the most powerful, professional programmer's tools ever published in a magazine are in this volume. Jim Hendrix's Small C compiler. Ed Ream's RED screen editor. A microcomputer subset of the Defense Department's official programming language, Ada. C and Forth and 68000 software. Because the magazine increased in size in 1983, this volume is bigger and better than ever.



Vol. 1 1976 (Item #013)

The material brought together in this volume chronicles the development in 1976 of Tiny BASIC as an alternative to the "finger blistering," front-panel, machine-language programming which was then the only way to do things. This is always pertinent for the bit crunching and byte saving, language design theory, home-brew computer construction and the technical history of personal computing.

Topics include: Tiny BASIC, the [very] first word on CP/M, Speech Synthesis, Floating Point Routines, Timer Routines, Building an IMSAI, and more.

Vol. 2 1977 (Item #014)

1977 found DDJ still on the forefront. These issues offer refinements of Tiny BASIC, plus then state-of-the-art utilities, the advent of PILOT for microcomputers and a great deal of material centering around the Intel 6800, including a complete operating system. Products just becoming available for reviews were the H-8, KIM-1, MITS BASIC, Poly Basic, and NIBL.

Articles are about Lawrence Livermore Lab's BASIC, Alpha Micro, String Handling, Cyphers, High Speed Interaction, I/O, Tiny Pilot & Turtle Graphics, many utilities, and even more.

Vol. 3 1978 (Item #015)

The microcomputer industry entered into its adolescence in 1978. This volume brings together the issues which began dealing with the 6502, with mass-market machines and languages to match. The authors began speaking more in terms of technique, rather than of specific implementations; because of this, they were able to continue laying the groundwork industry would

follow. These articles relate very closely to what is generally available today.

Languages covered in depth were SAM76, Pilot, Pascal, and Lisp, in addition to RAM Testers, S-100 Bus Standard Proposal, Disassemblers, Editors, and much, much more.

Vol. 4 1979 (Item #016)

This volume heralds a wider interest in telecommunications, in algorithms, and in faster, more powerful utilities and languages, innovation is still present in every page, and more attention is paid to the best ways to use the processors which have proven longevity—primarily the 8080/8085, 6502, and 6800. The subject matter is invaluable both as a learning tool and as a frequent source of reference.

Main subjects include: Programming Problems/Solutions, Pascal, Information Network Proposal, Floating Point Arithmetic, 8-bit to 16-bit Conversion, Pseudo-random Sequences, and Interfacing a Micro to a Mainframe—more than ever!

Vol. 5 1980 (Item #017)

All the ground-breaking issues from 1980 in one volume! Systems software reached a new level with the advent of CP/M, chronicled herein by Gary Kildall and others (DDJ's all-CP/M issue sold out within weeks of publication). Software portability became a subject of greater import, and DDJ published Ron Cain's immediately famous Small-C compiler—reprinted here in full. Contents include: The Evolution of CP/M, a CP/M-Flavored C Interpreter, Ron Cain's C Compiler

for the 8080. Further with Tiny BASIC, a Syntax-Oriented Compiler Writing Language, CP/M to UCSD Pascal File Conversion, Run-time Library for the Small-C Compiler and, as always, even more!

Vol. 6 1981 (Item #018)

1981 saw our first all-FORTH issue (now sold out), along with continuing coverage of CP/M, small-C, telecommunications, and new languages. Dave Cortesi opened "Dr. Dobb's Clinic" in 1981, beginning one of the magazine's most popular features.

Highlights: Information on PCNET, the Conference Tree, and The Electric Phone Book, writing your own compiler, a systems programming language, and Tiny BASIC for the 6809.

Vol. 7 1982 (Item #019)

In 1982 we introduced several significant pieces of software, including the RED text editor and the Runic extensible compiler, and we continue to publish utility programs and useful algorithms. Two new columns, The CP/M Exchange and The 16-Bit Software Toolbox, were launched, and we devoted special issues to FORTH and telecommunications. Resident Intern Dave Cortesi supplied a year of "Clinic" columns while delivering his famous review of JRT Pascal and writing the first serious technical comparison of CP/M-86 and MSDOS. This was also the year we began looking forward to today's generation of microprocessors and operating systems, publishing software for the Motorola 68000 and the Zilog Z8000 as well as Unix code. And in December, we looked beyond, in the provocative essay, "Fifth-generation Computers."

Complete your reference library. Buy the entire set of Dr. Dobb's Journals from 1976 through 1983, Bound Volumes 1-8, for \$195.00. That's \$34.00 off the combined individual prices—a savings of almost 15%!

To order by credit card, call toll free: 1-800-528-6050 ext. 4001. Refer to the item number of each volume. Or mail this coupon to: Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303.

YES! ☐ Please send me the following Volumes of **Dr. Dobb's Journal**.

Payment must accompany your order.

Please charge my: ☐ Visa ☐ MasterCard ☐ American Express

I enclose ☐ Check/money order

Card # _____ Expiration Date _____

Signature _____

Name _____ Address _____

(please, no P.O. Boxes)

City _____ State _____ Zip _____

Mail to Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303

Vol. 1	_____ x	\$27.75 =	_____
Vol. 2	_____ x	\$27.75 =	_____
Vol. 3	_____ x	\$27.75 =	_____
Vol. 4	_____ x	\$27.75 =	_____
Vol. 5	_____ x	\$27.75 =	_____
Vol. 6	_____ x	\$27.75 =	_____
Vol. 7	_____ x	\$30.75 =	_____
Vol. 8	_____ x	\$31.75 =	_____
All 8	_____ x	\$195.00 =	_____
		Sub-total \$	_____

California residents add applicable sales tax _____%

Postage & Handling Must be Included with order.

Please add \$1.75 per book in U.S. (\$4.25 each surface mail outside U.S. Foreign Airmail rates available on request.)

TOTAL \$ _____

Listing Three

OUTPUT FROM ZIKO TO CONSOLE DURING TEST OVERLAY CONSTRUCTION

```
->CMND(CTEST);
CMND(
CTEST)<--Command File Input
ROOT(
MAIN)
;
OVINIT,<--Implied Include List
OVLAY,
L0;
END;<--Searching For Required Modules

-- Building File on Disk --

Loading      (0:OVINIT .REL)
Loading      (0:OVLAY .REL)
Loading      (0:L0 .REL)
Loading from(0:CRUNLIB .REL)

-- Filename --          -- File Information --
0:MAIN      .COM        P.Base: 0100      P.Size: 0EA7      Seg.End+1: 1182
                  D.Base: 0FBB      D.Size: 01C7
                  C.Base: 1182      C.Size: 0000

SYMFIL(
CTEST)<--Creating Symbol File
;
OVL(
1,
LIA)
;
INCL(
LIA)
```

(Continued on next page)

Now available with 8087 Support!

MTBASIC

Basic Compiler

Features:

Multi-line functions	Multitasking
No runtime fee	Windowing
Handles interrupts	Interactive
Fast native code	Compiles in seconds

MTBASIC is easy to use since you can write programs in an interactive environment and then compile them using only one command. MTBASIC has many advanced features like multitasking, random file access, formatted I/O, assembly language calls, and ROMable code.

The MTBASIC package includes all the necessary software to run in interpreter or compiler mode, an installation program (so any system can use windows), demonstration programs, and a comprehensive manual.

Ordering

MTBASIC is available for CP/M, MS-DOS, and PC-DOS systems for \$49.95. MTBASIC with 8087 support is available for MS-DOS for \$79.95. Shipping is \$3.50 (\$10.00 overseas). MD residents add 5% sales tax. MC, Visa, checks and COD accepted.



P.O. Box 2412 Columbia, MD 21045-1412
301/792-8096

A Professional Quality Z80/8080/8085 Disassembler

WHEN YOU NEED SOURCE FOR YOUR CODE you need REVAS 3

REVAS interactively helps you:

- Analyse your software for modification
- disassemble files as large as 64K
- Assign Real labels in the disassembly
- Insert COMMENTS in the disassembly
- Generate a Cross Reference (XREF) listing

A 60 page manual shows how the powerful REVAS command set gives you instant control over I/O to files, printer, or console; how to do a disassembly; and even how the disassembler works! You get on line help, your choice of assembler mnemonics, control of data interpretation, and calculation in any number base!

REVAS runs in Z80 CPM computers; is available on 8" SSD (standard), RAINBOW, and other (ask) formats
Price: \$90.00 (plus applicable tax), Manual only: \$15.00

REVASCO

6032 Chariton Ave., Los Angeles, CA 90056
Voice: (213) 649-3575 Modem: (213) 670-9465

Listing Three

```

OLV(
1,
L1B)
;
INCL(
L1B)
;
END;!--Searching For Required Modules
;
-- Building File on Disk --
Loading      (0:L1B      .REL)
-- Filename --
0:L1B      .OVL      P.Base: 1182      P.Size: 000E      Seg.End+1: 11B7
                  D.Base: 119C      D.Size: 001B
                  C.Base: 11B7      C.Size: 0000

OVL(
1,
L1C)
;
INCL(
L1C)
;
END;!--Searching For Required Modules
;
-- Building File on Disk --
Loading      (0:L1C      .REL)
-- Filename --
0:L1C      .OVL      P.Base: 1182      P.Size: 00B2      Seg.End+1: 12F5
                  D.Base: 1240      D.Size: 00B5
                  C.Base: 12F5      C.Size: 0000

OVL(
2,
L2A)
;
INCL(
L2A)
;
END;!--Searching For Required Modules
;
-- Building File on Disk --
Loading      (0:L2A      .REL)
-- Filename --
0:L2A      .OVL      P.Base: 12F5      P.Size: 0045      Seg.End+1: 138C
                  D.Base: 1346      D.Size: 0046
                  C.Base: 138C      C.Size: 0000

DOVL;
--- Overlay Structure ---
Name      Number      P.Base      End      Parent
-----
L1A      0000      1182      11B6
L1B      0001      1182      11B6
L1C      0002      1182      12F4
L2A      0003      12F5      138B      L1C
DONE;
No Segment Under Construction

```

End Listings

WHO WOULD DIG THROUGH 300 DISKS OF MSDOS SOFTWARE TO FILL 14 DISKS FULL OF GEMS?



\$12.00 STOCKING STUFFERS YOU'LL USE ALL YEAR!

DISK MS-1 ESSENTIAL UTILITIES

SWEEP is a powerful replacement for COPY. LU86403 makes (and unmakes) library files from your programs. (Bulletin boards contain many library files.) DSKRTN is a fancy disk utility which hides files, recovers files, and lists first sector of an erased file. WHEREIS finds file in subdirectories. SQUEEZE and UNSQUEEZE squeeze and expand files for faster and cheaper modem transfer. Plus WASH and several other file manipulators.

DISK MS-2 ASSEMBLER AND DISASSEMBLER

CHASM is compatible with the IBM macroassembler. ASMGEN is a disassembler whose output is compatible with the IBM macroassembler (works in 64K). MEMBRAIN converts extra RAM into a RAM disk. UNWORDSTAR converts Wordstar files to ASCII and vice versa. FSPPOOL is a printer output to file redirector and SDIR is a super directory program. (And more...)

DISK MS-3 ADVENTURE

This is the original text-style adventure game that has kept whole throngs of (otherwise) serious programmers glued to their screens. It's ready to run but we also include source in CII and notes about how to expand this adventure (or create your own). Great fun for all computer users.

DISK MS-4 SINGLE USER RBBS

Turn your PC (or compatible) into a bulletin board and get called by the world. Includes source.

DISK MS-5 FILE AND DIRECTORY UTILITIES

UTIL sorts directories, moves files between directories, redefines the keyboard, and views files. Z sorts files. VTYPE is an easier to use TYPE. ST pages backwards or forwards through files. PROTECT AND UNPROTECT prevents accidental (or purposeful) erasure of files or programs. PCWINDOW a resident multi-function routine that handles notepads, timing, ASCII translation etc. Very handy Sidekick-like program.

DISK MS-6 FIG FORTH

(Includes MS-7, FORTH Source)

DISK MS-8 THREE GAMES

CASTLE is a graphic and conversational adventure. TREKRUN is a startrek style game. BOUNCING BABY is a finger exercise that will enthrall a 10-year-old.

DISK MS-9 PASCAL GAMES, PC MUSICIAN, CALCULATOR

This is our educational disk. PC MUSICIAN lets you write songs right on the screen and test them as you go. "I amazed my friends when I sat down at my computer and played..." The PASCAL GAMES aren't all that exciting, but the source should give you a good start on creating your own. PROGRAMMER'S CALCULATOR shows you how the stack, memory, and CPU registers work, plus it's highly calculating.

DISK MS-10 MODEM SOFTWARE

If you want to talk to bulletin boards this is your spot. QMODEM is the fanciest of the MODEM7 packages we've seen. It's powerful but easy to use. KERMIT is the standard package for communicating with mainframe (there are versions of KERMIT for nearly all the mainframes).

DISK MS-11 MULTI-USER RBBS 12.3 SYSTEM

Another complete RBBS, this one allows more than one user at a time.

DISK MS-12 SOURCE OF RBBS 12.3 SYSTEM

DISK MS-13 TEXT EDITOR & FORMATTER

ROFF4 is UNIX-like text formatter, PC-Write is one of the most famous of the freeware editors.

DISK MS-14 FORTH 84

For Super Fast Service Call
503-382-5060

MICRO CORNUCOPIA

The Micro Technical Journal

Disks \$12.00 each Postpaid
MICRO CORNUCOPIA
P.O. BOX 223
BEND, OREGON 97709



**"All you CPM people
who wanted to move
to IBM, but couldn't...
now you can — and
bring your CPM
software with you!"**

From
GTEK, the
leader in devel-
opment hard-
ware and
software,
comes the
CPM
user's
dream.

The
majority
of advances
being made
in the computer industry today are
being made for PC-DOS and MS-DOS machines.

If you are a CPM user who wanted to convert
to these operating systems but didn't because you
had to abandon or rewrite your CPM programs,
now you can make your move without losing or
rewriting a single
program!

The dream from
GTEK is a package
that makes your
PC-DOS (IBM PC,
XT, AT) or MS-DOS
(IBM clones like
COMPAQ, Leading
Edge, Televideo)
machines think
they're being run by
an 8080 or Z80 microprocessor with CPM!

The GTEK package contains a CPEmulator™
and CPM Conversion Utility that allows you to
read and write popular CPM disk formats such as
Kaypro and Osborne.

How easy is it to use? Two steps. (1) Use the
Conversion Utility to copy the CPM program to
an MS-DOS or PC-DOS diskette. (2) Use the Bind
Utility to attach the CPEmulator to your program.
And that's it. It's ready to run.

You won't see any difference when you run
the CPM program! Now all the work you did in
CPM is still valuable when you change to your
PC-DOS or MS-DOS machine.

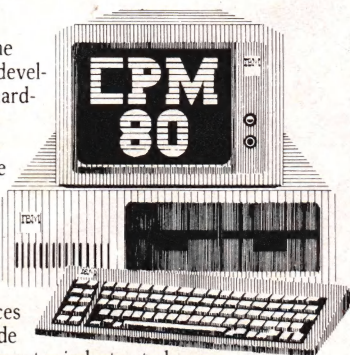
And here's the happy ending to the dream.
The CPEmulator is just \$199.

**Call GTEK's CPM Hotline...
1-601-467-9019**

GTEK, INC.

Development Hardware/Software
P.O. Box 289, Waveland, MS 39576
601/467-8048; telex 315-814 (GTEK UD)

*Registered trademarks. GTEK, CPM, PC-DOS,
MS-DOS, IBM PC, XT, AT, COMPAQ, Leading
Edge, and CPEmulator are also registered
trademarks.



**If you've been using
CPM on any of these
machines you can now
move to PC-DOS or
MS-DOS operating
systems.**

Kaypro • Osborne • Heath/
Zenith • Radio Shack •
Epson • DEC • IBM • Mor-
row • Xerox • HP • Sanyo
• Intertec • Cromenco •
Televideo • Compupro •

ADVERTISER INDEX

Reader Service No.	Advertiser	Page No.	Reader Service No.	Advertiser	Page No.
158	AMPRO Computers, Inc.	112	64	Micro Cornucopia	127
176	Advent Products, Inc.	64-65	110	Micro Interface Corp.	93
121	Arity Corporation	47	*	Micromint	26
123	BD Software	79	106	Micro Solutions	12
159	Blaise Computing	111	*	MicroMethods, Inc.	90
161	Borland International	C-4	105	MicroProcessors Unlimited	101
126	Boston Software Works	83	103	MicroSmith	101
177	Byte Works, Inc.	94	136	Microcomputer Systems Conslts.	79
63	C User's Group	101	*	Microtec Research	76
*	C Ware Corporation	77	71	Mitek	33
178	Chalcedony	15	*	Mix Software	67
81	Cogitate, Inc.	101	128	Morgan Computing Company	37
122	CompuView	45	79	Mystic Canyon Software	27
127	Computer Control Systems	95	168	NCI	35
129	Computer Helper Industries, Inc.	100	72	Northwest Computer Algorithms	107
96	Computer Innovations	113	137	OES Systems	101
82	Creative Programming	104	124	Optotech	1
*	DDJ Bound Volume	124	73	Overland Data, Inc.	104
*	DDJ C Products	108-109	76	Personal Tex Inc.	97
*	DDJ Sourcebook	122	91	Phoenix Computer Products	C-3
*	DDJ Subscription	120	139	Phoenix Computer Products	2-3
86	Dalsoft Systems	101	74	Plu Perfect Systems	86
130	Data Base Decisions	99	169	Poor Person Software	89
162	Decamation	92	140	Productivity Products Int'l.	18
87	Digital Research Computers	69	143	Programmer's Shop	49
118	Diversified Educational Enterprises	58	141	Programmer's Shop	43
163	DRI	119	62	Prospero	61
179	Earth Computers	62	107	Quilt Computing	83
89	Ecosoft, Inc.	73	145	Rational Systems, Inc.	68
90	Edward K. Ream	77	170	Relational Database Systems	55
*	Emerald Systems Corp.	10	80	Revasco	125
92	Entelekon Software Systems	60	78	SLR Systems	97
188	Essential Software	36	171	Samkha	123
165	Everest Solutions	C-2	114	Seidl Computer Engineering	88
180	Executive Systems	75	85	SemiDisk Systems	81
93	Faircom	17	83	Soft Advances	91
94	Fox Software, Inc.	103	111	Soft Advances	91
95	GTEK	128	113	SoftCraft, Inc.	11
*	Gimpel Software	22	*	Soft Focus	88
*	Gimpel Software	105	88	Softaid, Inc.	125
97	Greenleaf Software, Inc.	71	115	Software Development Systems	101
98	Guidance Software	80	147	Solution Systems	118
131	Hallock Systems Consultants	20	151	Solution Systems	53
132	Harvard Softworks	24	148	Solution Systems	83
133	Hawaiian Software	63	149	Solution Systems	83
134	HiSoft	101	164	Spruce Technologies	51
99	Illyes Systems	91	142	Starlight Forth Systems	101
166	Inetco Systems	123	172	Sunny Hill Software	96
75	InfoPro Systems	92	144	Sunset Technology	93
*	Integral Quality	66	146	Symsoft	41
65	Intelliware, Inc.	100	104	Systems Management Assoc.	39
100	Kriya	57	173	TLM Systems	21
66	Laboratory Microsystems Inc.	77	174	TLM Systems	23
67	Lahey Computer Systems	83	175	TLM Systems	25
101	Lattice, Inc.	117	117	The Librarian	94
125	Lifeboat Associates	115	119	Tiny Tek, Inc.	97
135	Lugaru Software, Ltd.	56	150	Trio Systems	112
68	MacTutor	93	120	/usr/group - UniForum	121
69	Magnum Data, Inc.	118	77	UniPress Software	59
109	Manx Software	32	154	Vance Info Systems	95
108	Manx Software	7	157	Vermont Creative Software	87
102	Mark Williams	13	112	Wendin, Inc.	9
70	Martian Technologies	93	156	Whitesmith Ltd.	31
167	Martin Scot Dev. Corp.	90	116	Wizard Systems	89
84	Megamax, Inc.	95	160	Zedcor	99

* This advertiser prefers to be contacted directly: see ad for phone number.

Advertising Sales Offices

East Coast

Walter Andrzejewski (617) 567-8361

Midwest

Michele Beaty (317) 875-8093

Northern California/Northwest

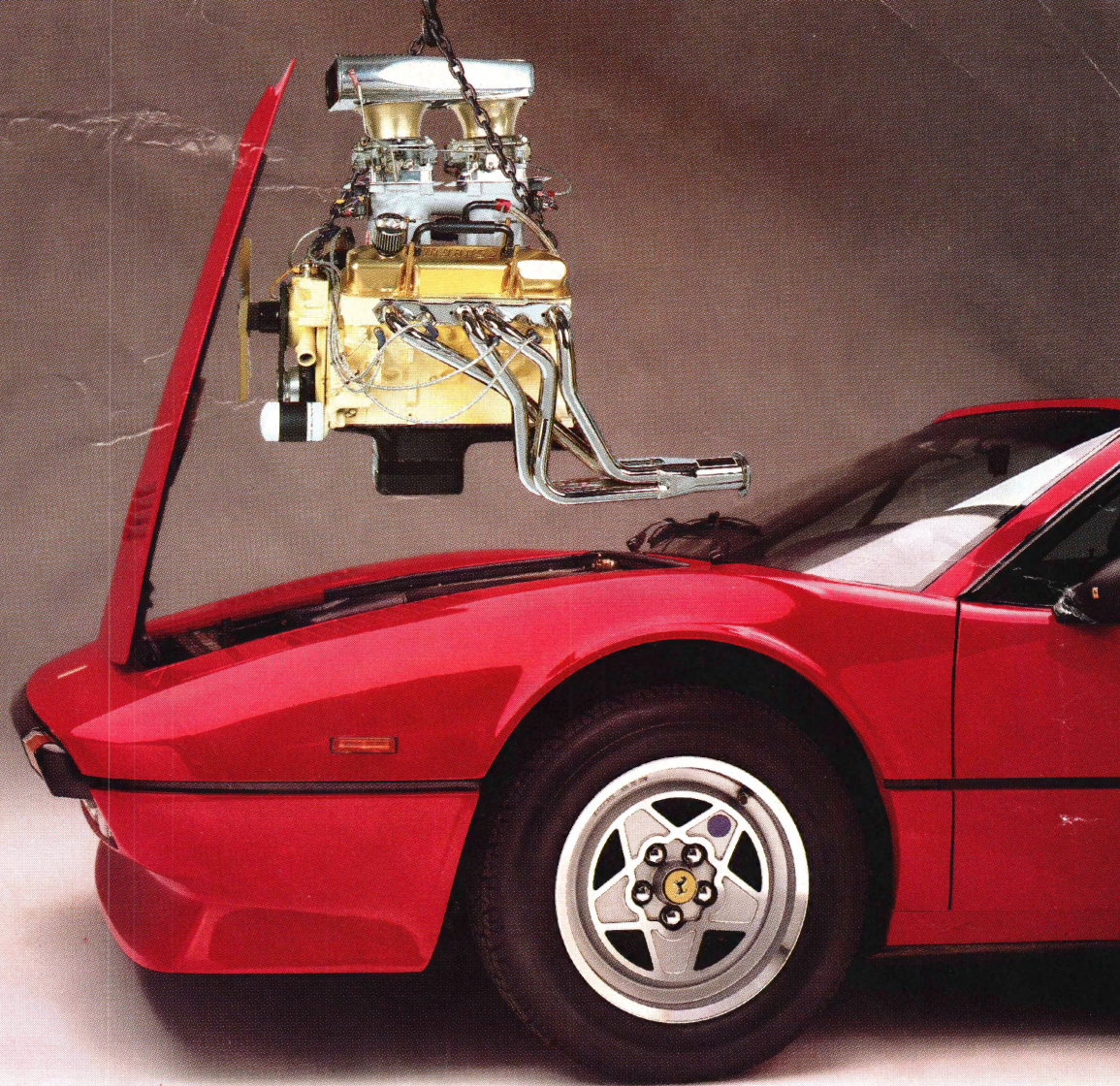
Lisa Boudreau (415) 424-0600

Southern California/AZ/NM

Beth Dudas (714) 643-9439

Advertising Director

Shawn Horst (415) 424-0600



AT™ Pfantasies for your PC or XT.™

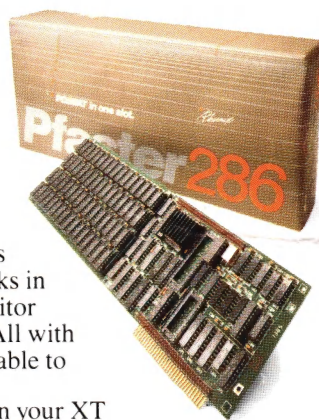
Want better speed and memory on your PC or XT without buying an AT?

You've got it!

Phoenix's new Pfaster™286 co-processor board turns your PC or XT into a high-speed engine 60 percent faster than an AT. Three times faster than an XT. It even supports PCs with third-party hard disks. But that's only the beginning.

You can handle spreadsheets and programs you never thought possible. Set up RAM disks in both 8088 and 80286 memory for linkage editor overlays or super-high-speed disk caching. All with Pfaster286's 1mb of standard RAM, expandable to 2mb, and dual-mode design.

You can develop 8086/186/286 software on your XT faster. Execute 95 percent of the application packages that run on the AT, excluding those that require fancy I/O capabilities your PC or XT hardware just isn't designed to handle. Queue multi-copy, multi-format print jobs for spooling. Or, switch to native 8088 mode to handle



hardware-dependent programs and back again without rebooting. All with Pfaster286's compatible ROM software. And, Pfaster286 does the job unintrusively! No motherboard to exchange. No wires to solder. No chips to pull. Just plug it into a standard card slot, and type the magic word, "PFAST."

If you really didn't want an AT in the first place, just what it could do for you, call or write: Phoenix Computer Products Corp., 320 Norwood Park South, Norwood, MA 02062; (800) 344-7200. In Massachusetts, 617-762-5030.

Programmers' Pfantasies™
by

Phoenix

XT and AT are trademarks of International Business Machines Corporation. Pfaster286 and Programmers' Pfantasies are trademarks of Phoenix Computer Products Corporation. For the Ferrari aficionado: yes, we know this is a rear engine car. We are showing the addition of a second engine to symbolize how Pfaster can be added to your PC or XT to increase performance.

Circle no. 91 on reader service card.

SEE US AT FALL COMDEX

SAVE OVER 30% ON OUR GIFT PACKS!
60-DAY MONEY-BACK GUARANTEE

How Borland's Three New Holiday Packs Will Fill Your Stocking Without Emptying Your Piggybank.

Three special packs with dazzling discounts that will help get you into a Holiday mood. You can get some of Turbo, most of Turbo, or all of Turbo—including the two newest members of the Turbo family, Turbo GameWorks™ and Turbo Editor Toolbox™. You also get our unmatched 60-day money-back guarantee, quality products that aren't copy-protected.

TURBO NEW PACK \$95.00.

You get the two exciting new members of the Turbo Pascal family,

- TURBO GAMEWORKS, Chess, Bridge, and Go-Moku, complete with source code and a 200-page manual.
- TURBO EDITOR TOOLBOX, all the building blocks to make your own editors and word processors, complete with source code and a 200-page manual.

TURBO HOLIDAY PACK \$125.00.

You get all three of the Turbo family classics for only \$125.00 (about a 30% discount). Turbo Pascal 3.0 and Turbo Tutor and Turbo DataBase Toolbox—all for just \$125.00.

- TURBO PASCAL combines the fastest Pascal compiler with an integrated development environment.
- TURBO TUTOR teaches you step-by-step how to use Turbo Pascal with commented source code for all program examples on diskette.
- TURBO DATABASE TOOLBOX offers three problem-solving modules for your Turbo Pascal programs: Turbo Access, Turbo Sort, and GINST, which generates a ready-to-run installation program that lets you forget about adapting your software to specific terminals.

TURBO HOLIDAY JUMBO PACK \$245.00.

This is it—the whole thing, the entire Turbo family including its two newest members. You get:

- Turbo Pascal
- Turbo Graphix Toolbox
- Turbo Tutor
- Turbo DataBase Toolbox
- Turbo GameWorks
- Turbo Editor Toolbox

and you pay only \$245.00 for all six! Which means that you're getting everything at only about \$40 a piece. Quite a holiday deal. (And if you already own one or several members of the Turbo family, be creative—nothing can stop you from buying the Jumbo Pack, picking out the ones you already have and giving the rest as holiday gifts to family and friends. At these prices you can afford to give to others and to yourself.) Speaking of Holidays, this offer lasts until March 31, 1986. (At Borland, we like to make the Holidays last.)



4585 Scotts Valley Drive, Scotts Valley CA 95066
Phone (408) 438-8400 Telex 172373

Copyright 1985 Borland International BI-1017

Turbo Pascal and Turbo Tutor are registered trademarks and Turbo DataBase Toolbox, Turbo Graphix Toolbox, Turbo Editor Toolbox, Turbo GameWorks, and MicroStar are trademarks of Borland International, Inc. WordStar is a trademark of MicroPro International Corp. Multi-Mate is a trademark of Multi-Mate International Corp. Microsoft is a registered trademark and Word is a trademark of Microsoft Corp. WordPerfect is a trademark of Satellite Software International.



NEW! TURBO GAMEWORKS \$69.95.

Our new Turbo GameWorks offers games you can play and replay without Turbo Pascal or revise and rewrite with Turbo Pascal 3.0. We give you the source code, the manual, the diskettes and the competitive edge. Chess, Bridge and Go-Moku. State-of-the-art games that let you be player, referee, and rules committee all at once because you have the Turbo Pascal source code. Learn exactly how the games are made—so you can go off and make your own. And Turbo GameWorks is the only quality game you can buy that is not copy-protected. Sold separately, only \$69.95. (Just \$47.50 if you buy the Turbo New Pack.)

NEW! TURBO EDITOR TOOLBOX \$69.95.

Build your own word processor—for only \$69.95!

You get ready-to-compile source code, a full-featured word processor that looks and acts like WordStar™, and a 200-page manual that tells you how to integrate the editor procedures and functions into your programs. With Turbo Editor Toolbox you can have the best of all word processors. You can make WordStar behave like Multi-Mate. Support windows just like Microsoft's Word. And do it as fast as WordPerfect does it. Incorporate your new "hybrids" into your programs to achieve incredible control and power. Sold separately, only \$69.95. (If you buy the Turbo New Pack, the price drops to just \$47.50.)

Holiday Gift Packs, Turbo GameWorks™ & Turbo Editor Toolbox™

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133.

The holiday packs include an upgrade coupon for both options so you get BCD and 8087 support for \$39.95 (regularly \$55.00).

Carefully describe your computer system!

Mine is: ☐ 8-bit ☐ 16-bit
I use: ☐ PC-DOS ☐ MS-DOS
☐ CP/M-80 ☐ CP/M-86

My computer's name and model is: _____

The disk size I use is: ☐ 3 1/2" ☐ 5 1/4" ☐ 8"

Name: _____
Shipping Address: _____
City: _____ Zip: _____
State: _____ Telephone: _____

Quantity

* Turbo Holiday Jumbo Pack	\$245.00
* Turbo Holiday Pack	\$125.00
* Turbo New Pack	\$95.00
Pascal	\$69.95
Pascal w/8087	\$109.90
Pascal w/BCD	\$109.90
Pascal w/8087 and BCD	\$124.95
Turbo DataBase	\$54.95
Turbo Graphix	\$54.95
Turbo Tutor	\$34.95
Turbo Editor	\$69.95
Turbo GameWorks	\$69.95

These prices include shipping to all U.S. cities. All foreign orders add \$10 per product ordered.

Amount: (CA add 6% tax) _____
Payment: ☐ VISA ☐ MC ☐ Check ☐ Bank Draft
Credit Card Expiration Date: _____ / _____
Card # _____

***Gift Pack
Offers Last
Until March 31,
1986**

H11

NOTE: Turbo Editor Toolbox and Turbo GameWorks are available for the IBM PC and true-compatibles using Turbo Pascal 3.0 ONLY.